Contract No. N62269-91-C-0206
Report No. NADC-91123-60

APPLICATION OF AN ADAPTIVE CLUSTERING NETWORK TO FLIGHT CONTROL OF A FIGHTER AIRCRAFT

James C. Smith, Ph.D.
Robert V. Walters (STR Corp.)
Henry R. Jex
Bimal L. Alponso
SYSTEMS TECHNOLOGY INC.
13766 S. Hawthorne Blvd
Hawthorne CA, 90250

19 DECEMBER 1991

FINAL REPORT

Approved for public release; distribution is unlimited

Prepared for
Air Vehicles and Crew Systems Technology Department (Code 6012)
NAVAL AIR DEVELOPMENT CENTER
Warminster, PA 18974-5000

NOTICES

REPORT NUMBERING SYSTEM — The numbering of technical project reports issued by the Naval Air Warfare Center, Aircraft Division, Warminster is arranged for specific identification purposes. Each number consists of the Center acronym, the calendar year in which the number was assigned, the sequence number of the report within the specific calendar year, and the official 2-digit correspondence code of the Functional Department responsible for the report. For example: Report No. NAWCADWAR-92001-60 indicates the first Center report for the year 1992 and prepared by the Air Vehicle and Crew Systems Technology Department. The numerical codes are as follows:

CODE	OFFICE OR DEPARTMENT
00	Commanding Officer, NAWCADWAR
01	Technical Director, NAWCADWAR
05	Computer Department
10	AntiSubmarine Warfare Systems Department
20	Tactical Air Systems Department
30	Warfare Systems Analysis Department
50	Mission Avionics Technology Department
60	Air Vehicle & Crew Systems Technology Department
70	Systems & Software Technology Department
80	Engineering Support Group
90	Test & Evaluation Group

PRODUCT ENDORSEMENT — The discussion or instructions concerning commercial products herein do not constitute an endorsement by the Government nor do they convey or imply the license or right to use such products.

Reviewed By: Robert F. Williams Date: 5/18/92
Contracting Officer's Technical Representative (COTR)
or
Point Of Contact (POC)

Reviewed By: [signature] Date: 5/18/92
Branch Head

Reviewed By: [signature] Date: 5/20/92
Division Head

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	19 December 1991	SBIR Phase I Final Report

4. TITLE AND SUBTITLE
Application of an Adaptive Clustering Network to Flight Control of a Fighter Aircraft

5. FUNDING NUMBERS
C: N62269-91-C-0206

6. AUTHOR(S)
James C. Smith
Robert V. Walters
Henry R. Jex
Bimal L. Aponso

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Systems Technology, Inc.
13766 S. Hawthorne Blvd.
Hawthorne, CA 90250

STR Corporation
10700 Parkridge
Reston, Virginia 22091

8. PERFORMING ORGANIZATION REPORT NUMBER
STI TR 1285-1

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Vehicle and Crew Systems Technology Department
(Code 6012)
NAVAL AIR DEVELOPMENT CENTER
Warminster, PA 18974

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NADC-91123-60

11. SUPPLEMENTARY NOTES
Technical Monitor: Robert D. Digirolamo (Code 6012)

12a. DISTRIBUTION/AVAILABILITY STATEMENT
A. Approved for Public Release; Distribution is Unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

An artificial neural net controller was developed for a typical fighter aircraft's longitudinal Stability and Control Augmentation System (SCAS), using elevator and thrust-vector-angle controls, and operating at high angle of attack. The "baseline" neurocontroller (NC) was in the feedforward loop (Kawato Type-C), with inputs from the pilot's pitch rate commands and rates and SCAS feedback error. An Adaptive Clustering Network algorithm was used to train the radial-basis-function neurons. Significant improvements in performance resulted from the NC action and these effects were analyzed by frequency domain describing functions. Thrust vector failures were handled satisfactorily, but reconfiguration of the SCAS was not possible within the simplified aircraft and NC simulation. Emphasis was on validation methods and ways to interpret the time-varying non-linear NC effects. Phase II recommendations are included, such as: ways to choose signals for the neural net to more efficiently identify and separate failures of correlated control effectors; the further use of frequency domain describing functions to identity neurocontroller dynamic processes; and the development of a Neuro-controller Analysis Toolbox with diagnostic forcing functions, methods, analyses, and benchmark

14. SUBJECT TERMS		15. NUMBER OF PAGES
Artificial Neural Nets	Adaptive Clustering Network	68
Aircraft Flight Control	Nonlinear Control Analysis	16. PRICE CODE
Automatic Controls	Kawato Type-C Controller	

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

13. ABSTRACT (Continued)

criteria for evaluation to a common NC standard.

PREFACE

This Phase I SBIR project was performed by Systems Technology, Inc. of Hawthorne, CA, under the sponsorship of the Naval Air Development Center, Flight Controls Group (Code 6012) at Warminster, PA. Technical monitor was Robert D. Digirolamo. James C. Smith, Ph.D., was the Principal Investigator at Systems Technology, Inc. Robert V. Walters was the Principal Investigator at STR Corporation of Reston, VA, who were one-third partners in applying their proprietary neural-net algorithms and simulation facilities. The work was performed during the mid-1991 period.

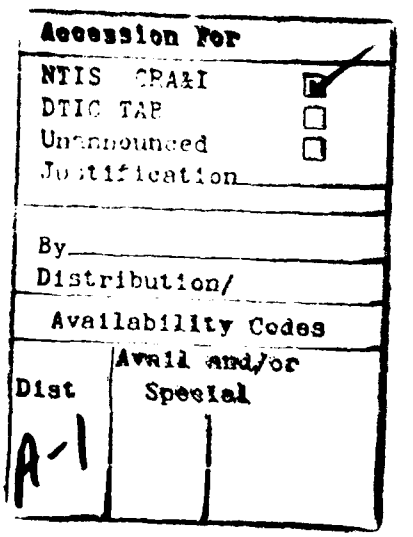


Table of Contents

List of Figures

List of Tables

I. INTRODUCTION

A. BACKGROUND AND MOTIVATION

The need for on-line adaptation, the capacity to deal with essential nonlinearities and rapid real-time response to damage has prompted the search for control strategies for application in advanced high-bandwidth aircraft control. Among the strategies for control system design which have been explored over the past 20 years, are a number of implementations which offer real-time adaptive control and/or self reconfiguration. Recently, such systems based on modular adaptive elements and widely referred to as Artificial Neural Network (ANN) technology have emerged as promising candidates for adaptive control design. Conventional control systems must be specifically designed to tolerate and compensate for known design properties of the object system. Controllers based on ANN technology, however, have the ability to compensate for system behavior unknown at design time by *learning* to attain some specific control objective in real time. The potential for application of ANN technology to aircraft control has burgeoned with the proliferation of digitally based flight controllers, owing largely to the flexibility of the hardware and its generic capacity for reconfigurability in real time. The present study, while intended to demonstrate that the use of ANNs in control system architecture can provide certain benefits in terms of safety and flexibility, has proven additionally beneficial in elucidating the use of analytical procedures for identifying and evaluating ANN activity in real-time control context.

While a number of reports have been published on the application of neurocontrollers to feedback control of mechanical systems [Jordan, 1989; Kawato, 1990], there remains a significant amount of work to be done in order to properly assess and document the appropriateness of various types of ANN architectures to specific control problems. In accomplishing this, special emphasis must be placed on ascertaining whether the underlying neurodynamics are appropriate to the dynamics of the controlled element as well as the broad objectives of the control process [Barto, 1983; Bavarian, 1988; Coolidge, 1963; Guez, 1988; Holdaway, 1989].

Difficulties with detecting and reconfiguring of an aircraft to compensate for individual control effector failures (e.g., see Appendix A), led us to consider a different approach. This approach works within the constraints of an existing feedback loop architecture. It utilizes inputs and feedback error signals which are already in place to determine appropriate controlled-element responses to extant controls, by approximating the element's inverse in an open loop manner so as not to degrade closed loop stability. This strategy is analogous to the "Pursuit" mode of human operator behavior in the "Successive Organization of Perception" paradigm for an operator learning of a controlled element's dynamic response (i.e., Compensatory, Pursuit, and Precognitive) [Krendel and McRuer, 1960]. Eventually the skilled operator can respond mostly on the input (if it is displayed), with only vernier corrections from the error correcting "compensatory" loop. This approach, which simulates the human's ability to coordinate complex joint motions by inverting their complex impedances in a feedforward loop was successfully

applied by Kawato [1990]. After exploring alternative architectures, we finally adopted this particular scheme in view of its proven track record (in human neural nets and in Kawato's work) and its rich potential for aircraft control applications.

B. OBJECTIVES AND SCOPE

The primary overall objective of this project was to investigate and evaluate the use of artificial neural networks (ANN) in implementing flight control functions for advanced manned aircraft. In this regard, we have endeavored to take advantage of the known properties and limitations of aircraft control in conjunction with certain known attributes of a Stability and Control Augmentation System (SCAS) in order to provide a rational working environment for a real-time Neurocontroller (NC). This environment should, ideally, permit the NC to exercise sufficient authority to improve the basic control effectiveness under time-varying conditions and allow it to cope with unforseen failures of various aircraft control effectors while minimizing the impact of these failures upon aircraft safety and SCAS bandwidth. Although these objectives seem straightforward, a number of practical issues have tended to impede applications of neurocontrollers to high bandwidth aircraft control problems. These include: limited state measurement, difficult-to-identify plant dynamics, compromises between achieving good responses to commands while regulating against disturbances, fast-enough computations to catch sudden and serious (but controllable), failures, and "smart" enough to distinguish failure from normal manuevers.

Since a frontal assault on all of these issues was precluded by the budget and time limitations of the Phase I effort, a simplified set of objectives evolved:

i. Apply a feedforward-loop neural net controller to a practical but potentially challenging aircraft longitudinal control task, having one or two effectors.

ii. Demonstrate feasible fast-time operation of the STR Adaptive Clustering Network approach to mechanize a fast and efficient neural-net-controller.

iii. Evolve techniques and measures to reveal what the neurocontroller is doing to aid in its understanding and validation.

iv. Provide a workable simulation of the aircraft/SCAS/ANN NC system suitable for further expansion.

Some of the questions to be answered in Phase I included:

a. Is the simulation a valid representation of the selected aircraft/SCAS combination?

b. Can the ICMS architecture, as originally proposed, accomplish the reconfiguration and management characteristics as originally conceived or, if not, what is an appropriate adaptive architecture for conformance with known aircraft and mission objectives.

c. Can useful diagnostic methods be applied to understand and validate the selected ANN operation?

d. Does the ANN controller achieve improvements in basic control response?

e. Does the ANN controller achieve its principal design objective to rapidly cope with control failures?

f. What problems revealed in Phase I are to be addressed in Phase II?

C. APPROACH

Our approach to the problem addressed these basic issues:

a. Does the placement of the neural net in the control loop potentially satisfy the control and stability requirements for vehicle operation ?

b. Does the neural net adequately map the input space over its design locus ?

c Are the adaptation objectives of the controller achieved with the network architecture applied ?

d. Is the overall system performance adequately described by test inputs ?

The first of the above considerations was concerned with the placement of the adaptive elements in the control loop. In this regard, our intent was to place the adaptive element within the control structure in order to take the best advantage of the known "good" properties of neurocontrollers, and if at all possible, avoid some of the bad ones. We theorized that ANN adaptation should function well if required to map the inverse of a controlled element, much in the way a human operator or pilot would. The original proposed approach had suggested that we explore the application of a Jordan [1989] type of backpropagation-based ANN with classification type outputs intended only for the purpose of switching modes according to dynamically changing criteria as sensed by the adaptive element. In this sense the ANN would function as an intelligent arbiter of configuration state, thus relieving the pilot of configuration related activities as well as smoothing the transitions between states. Preliminary studies, however, indicated that, for several reasons, this architecture would not work well in the context of failed system detection/compensation. The Kawato Type-C [1990] feedback error approach was ultimately selected as a control loop configuration for the studies reported herein, but not before certain features of other network architectures in control context were revealed in pilot studies.

One of the primary reasons for the failure of our initial approach using forward modeling was the slowness of the backpropagation based element in converging on an appropriate solution. We felt that this design was unsuitable, especially at the relatively low frame rates which characterize some flight control systems. In seeking an alternative approach we became aware of two possibilities. One of these was the use of radial basis functions (RBFs) in constructing the network as reported in Musavi, et al, [1991]. This technique for performing rapid interpolations in high dimensional space, can potentially solve interpolative

mapping problems of input space much more efficiently than can backpropagation. This is due to the inherent ability of RBF processing elements (PEs) to adjust their receptive fields adaptively, thus "clustering" their influence around active regions of the input field. STR, early on, undertook to develop this concept, adapting it to yield continuous system outputs rather than discrete classification events. STR also endowed the RBF network architecture (RBFN) with the ability to instantiate new PEs when inputs occur beyond the receptive fields of extant elements. This architecture evolved into the STR proprietary paradigm, the ACN network (previously referred to as *adaptive-clustering interpolative mapping*, or ACIM network). STR's proprietary ACN network bears a functional resemblance to an *Adaptive Resonance Theory (ART)* network [Barto, 1983; Bavarian, 1988; Holdaway, 1984] with supervised learning. The ACN architecture has the following advantages for real time control:

- Self-organizing for enhanced feature determination
- Self-learning mechanism with fast initial learning rate
- Hardware implementation can allow real time processing
- Nearest neighbor classification of novelty
- Novelty detection and globally controllable arousal
- Self-scaling, and
- Self-stabilizing

ACNs represent a possible solution to the problem of maintaining adaptive potential in a near-stable environment - the so-called stability/plasticity dilemma. This plasticity is critical for any system which must continue to adapt without supervision. ACN is particularly well suited to the task of adapting to evolutionary process changes of the type likely to be experienced in the control context.

The availability, at STR Corp., of a rapidly responding clustering algorithm prompted us not only to re-examine the loop structure, but also to revise our strategy for the role of the ANN. In this regard, we adopted the strategy described in Section IV, using the ANN to exercise continuous control authority within constraints imposed by parallel operation of a conventional feedback controller. Further discussion of the rationale underlying this architecture will be found in Section IV.

An early observation of ACN performance influenced our choice of placing the ACN in the feedback error configuration [Kawato, et al, 1990]. As it became apparent that ANN architectures with rapid adaptation qualities were essential to the success of ANNs in flight control STR reviewed a number of alternatives to standard backpropagation including Counterpropagation, RBFs and Adaptive Resonance Technique (ART). In the course of these investigations, it was observed that rapidly adapting networks commonly exhibit extreme swings in output during the learning process. In a continuously learning system, it seemed prudent to adopt a control strategy which would minimize the influence of output extremes during the learning process, but allow the authority of the neurocontroller to increase as its input space became more fully mapped. The Kawato feedback error architecture [Kawato, 1990] appeared to fill this role, in that the ANN element operates in parallel with a conventional controller feedback controller; with

the system, in effect, responding to a summation of the two elements. This would tend to minimize the likelihood of extreme outputs from the neurocontroller influencing stability during early phases of input space acquisition.

Having thus selected a feedback structure, and a candidate network architecture, we proceeded to construct an appropriately simplified set of aircraft dynamics for evaluating the neurocontroller's performance. The aircraft dynamics are discussed in Section II. STR elected to utilize the Symantec THINK PASCAL environment on Apple MacIntosh computers for the development effort, due to their familiarity with the environment and the existence of ANN models and tools which could be adapted to the present problem. The actual simulations reported here were run at STI on a MacIntosh IIcx, and the time histories converted to MS-DOS format for frequency domain analysis, since appropriate tools are not presently available for the MacIntosh environment. Both the results and the methodology reported here form the basis for the work proposed for Phase II, which comprises the development of analytical tools specifically oriented toward validating and optimizing neurocontrollers in aircraft flight control systems.

II. SPECIFIC AIRCRAFT SELECTION

A. RATIONALE

An aircraft control situation was sought which would be challenging for a neurocontroller, yet capable of being simplified for Phase I, then expanded to full complexity in Phase II. The example case should also be easy to simulate on a Macintosh using the Symantec THINK PASCAL development environment, on which STR's ACN development shell had already been mechanized. It was also desirable that the failure detection example would not be a trivial problem for a simplified approach, e.g., a shaft-position monitor in a conventional control system.

The case chosen was the piloted longitudinal control of a fighter aircraft at high angle of attack (α) and low airspeed such as:

- **Execution of a flat or rolling "scissors" air combat maneuver (ACM)**
- **Low speed maneuvers due to ACM or other flight testing operations**

The "scissors" maneuver is accomplished to effect a deliberate and rapid loss of energy for arresting downrange travel and radically reducing turn radius. It can result in high α conditions at approximately 200-300 KIAS. Thrust vectoring, if available, will be important in vernier control of the aircraft. Operation at any high α state during ACM, or in thrust-vectoring development tests, is likely to be followed by a need to rapidly orient the aircraft to a gun target or an escape vector. To simplify this initial analysis, the aircraft is perturbed around a quasi-steady trim condition.

We have assumed, here, that the aircraft's controls are: the all-moving stabilizer/elevator operated by a fast servo actuator; and thrust vector deflection by a pair of exhaust deflection paddles operated by a pair of slower servo actuators. At the 220 kt airspeed and 30° angle-of-attack trim condition, the inherent pitch stability is negligible, and the pilot uses the fly-by-wire pitch control stick to maintain the aircraft line of sight via a Pitch-Rate-Command Attitude-Hold Mode of the aircraft's Stability and Control Augmentation System (SCAS) controller. Because the thrust must be approximately half of the weight in this low airspeed condition, thrust vector control provides more pitching effectiveness than the elevator, thereby requiring both control surfaces to achieve the needed aircraft pitching motions. The pilot must employ fairly aggressive pitch rate commands because the aircraft is sluggish and the aircraft disturbances due to buffet and target motions are all of moderate bandwidth.

In this case, the most likely failure mode is gradual or sudden burnoff of the thrust paddles, which would not be detectable by a shaft position monitor. Another failure mode could be loss of part or all of the stabilizer by battle damage or a "hardover" failure within the limited authority SCAS. These types of failure alter the apparent response properties of the "plant" (servos plus aircraft) in ways that are difficult to separately detect, because both the elevator and thrust vector deflection occur near the tail.

If both were in identical axial locations, had identical servo dynamics, and were coupled to share the task, then it would be impossible to segregate their effects, because both sets of inputs and responses would be correlated identically. However, the thrust paddles' effective center-of-pressure is behind the elevator's effective center-of-pressure, so some subtle differences between the ratios of normal acceleration and pitch acceleration are present. These could be detected by a neural-net monitoring suitable sensors, and the control scheme could be reconfigured appropriately. Appendix A contains a brief analysis of this approach, using the dynamic insights gained from past experience to suggest the best set of sensors for an efficient neural-net thrust-vector failure detector. [Jex, 1991]. After considerable analysis, described more fully in Appendix A, it was decided to substitute the feedforward controller example for the failure-detection example, leaving the latter as a separate issue for the Phase II development.

B. VEHICLE DYNAMICS AND SIMULATION

1. Aircraft Aerodynamic Model

In the course of ongoing controls research, STI has utilized and developed a variety of aircraft models which function in a range of simulation environments. For this problem, we selected an STI simplification of a fully implemented non-linear 6 DOF simulation of a twin-tailed fighter-attack aircraft with thrust vectoring. The STI model was validated in experiments conducted by STI and was implemented in the SYSL language. It has been used in recent research in the area of task-tailored flying qualities. The relevant physical characteristics of this aircraft are summarized in Appendix A (Table 1).

As noted earlier, the selected trim condition was 220 kts IAS, at $\alpha = 30°$. At these conditions the lift-curve slope, C_{L_α} is near zero, the thrust/weight ratio is near T/W = .50, and the pitching moment curve slope, C_{m_α} ranges from slightly stable to slightly unstable, depending on the C.G.

2. Low Order Equivalent System Model

Because the pitch static stability at this large angle-of-attack is small or negative, it can be assumed to be zero for Phase 1, thereby simplifying the aircraft equations of motion to only the pitch attitude degree-of-freedom. The Low Order Equivalent System (LOES) transfer function model for the aircraft response to controls is, then:

$$\frac{Q}{STAB} = \frac{M_{\delta_{STAB}}}{(s - M_q)}; \qquad \frac{Q}{VEC} = \frac{M_{\delta_{VEC}}}{(s - M_q)} \qquad \left[\frac{\text{deg/sec}}{\text{deg}}\right] \qquad \text{(1a, 1b)}$$

Here:

$$M_q = -.31\,\text{sec}^{-1}$$
$$M_{\delta_{STAB}} = -1.23\,\text{sec}^{-2}$$
$$M_{\delta_{VEC}} = -2.53\,\text{sec}^{-2}$$

(The negative signs on the M_δ coefficients arise from the definition of $+\delta$ giving +lift.)

To further simplify the simulation, it was assumed that the pilot was commanding pitch rate to hold the pitch near 30° (i.e., to track an overhead target line-of-sight) via a pitch-rate-attitude-hold fly-by-wire SCAS mode. Only this inner loop was simulated, as this is where the neurocontroller was to be located.

A typical fast servo for the STAB was included (time constant of .032 seconds).

$$\frac{STAB}{STAB_{CMD}} = \frac{1}{T_{STAB} + 1} = \frac{1}{.032s + 1} \tag{2a}$$

The thrust paddles require a slower, .12 second time-constant servo:

$$\frac{VEC}{VEC_{CMD}} = \frac{1}{T_{VEC}s + 1} = \frac{1}{.12s + 1} = \frac{8.33}{s + 8.33} \tag{2b}$$

The simulation and controller were sampled at 20 Hz (T_s = .050 sec), which is a LOES representation of the cumulative delays of the SCAS sensor/computer ensemble.

3. Block Diagram

The resulting Phase I block diagram is given in Figure 1. Notice that the neurocontroller (NC) block acts as a feedforward from the command pitch rate (and its derived acceleration), creating a so-called "Pursuit Control Architecture" in the sense of McRuer and Krendel [Krendel and McRuer, 1960]. The rationale behind this NC concept is discussed in Sections I-A and IV-A.

For Phase I the pitch control moments from the STAB and VEC deflections were roughly equalized. VEC deflection is nearly 2 times as effective as STAB deflection. Here:

$$\delta_{VEC\ CMD} = -1.0\ K_Q Q_E \qquad [deg] \tag{3a}$$

$$\delta_{STAB\ CMD} = -1.6\ K_Q Q_E \qquad [deg] \tag{3b}$$

The minus sign accounts for the negative sign definition of $M_{\delta STAB}$ and $M_{\delta VEC}$. For most of the cases shown here, K_Q = 1.0 deg per deg/sec.

4. Transfer Functions Relations

The transfer functions between the various signals in the closed-loop system are based on the fundamental feedback loop relationships given below:

a. Controlled Elements:

$$Y_{C\ STAB} = \frac{Q}{STAB_{CMD}} = \left(\frac{-1.0}{T_{STAB}s + 1}\right)\left(\frac{M_{\delta\ STAB}}{s - M_q}\right) \qquad [sec] \tag{4a}$$

$$Y_{C\ VEC} = \frac{Q}{VEC_{CMD}} = \left(\frac{-1.0}{T_{VEC}s + 1}\right)\left(\frac{M_{\delta\ VEC}}{s - M_q}\right) \qquad [sec] \tag{4b}$$

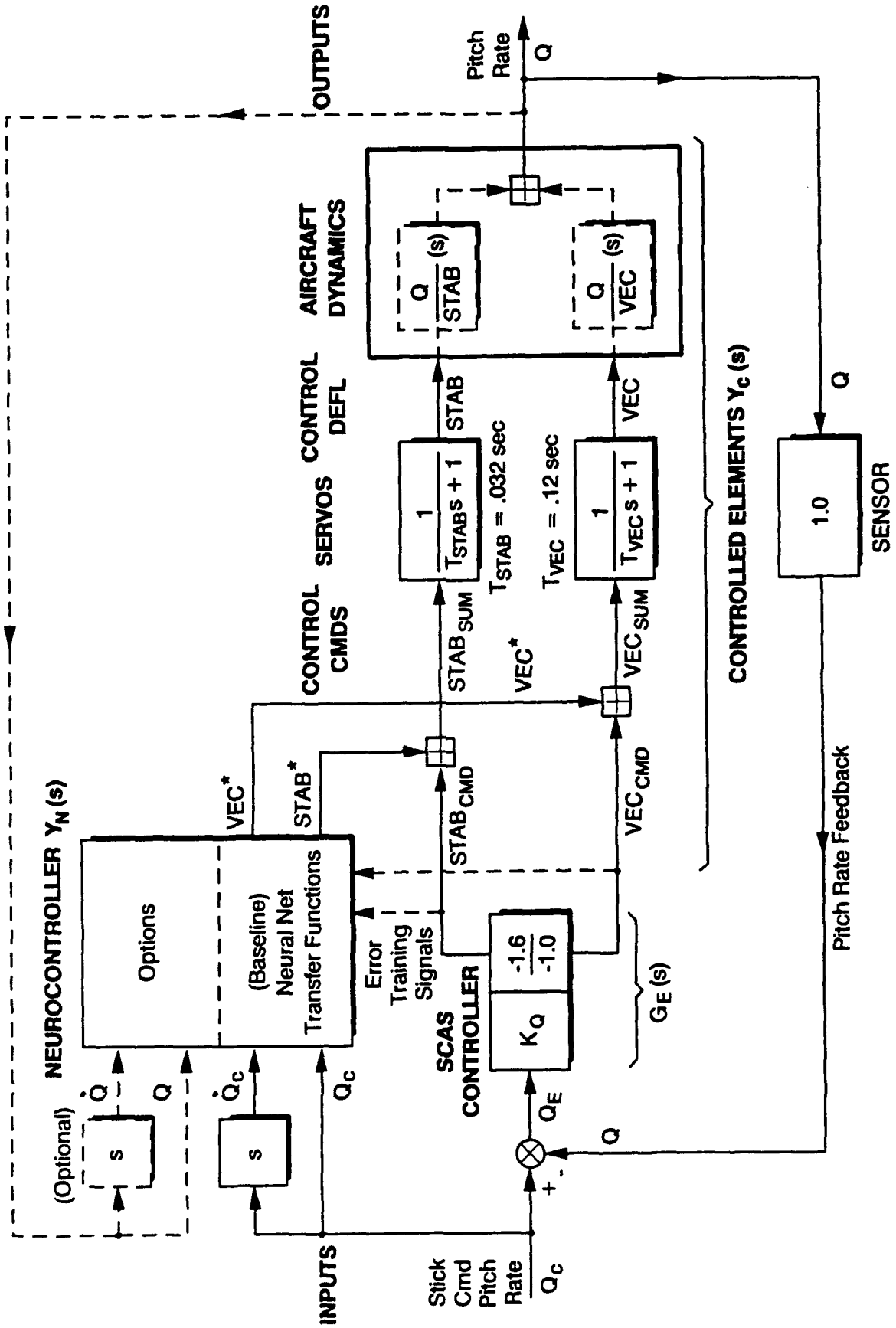


Figure 1. System Block Diagram

b. Control Laws

Feedback Error Controller (Pure Gain)

Let C_E be the reference control command from the error feedback (here ref. $C_E = VEC_{CMD}$)

$$G_E = \frac{C_E}{Q_E} \qquad \left[\frac{\text{deg}}{\text{deg/sec}}\right]$$
$$= STAB_{CMD} = 1.6K_Q Q_E \qquad (5)$$
$$= VEC_{CMD} = K_Q Q_E$$

The 1.6 factor was used to better balance the control deflections.

Neural Net Controller (Feedforward Loop on Cmd and Cmd Rate)

$$Y_N = \frac{C^*}{Q_C} = f\left(Q_C, \dot{Q}_C\right) \qquad \left[\frac{\text{deg}}{\text{deg/sec}}\right] = [\text{sec}] \qquad (6)$$

c. Open/Closed Loop Relations

Opened Loop

$$Y_{OL} = \frac{Q}{Q_E} = \frac{(G_E + Y_N)Y_C}{1 - Y_N Y_C} \qquad \left[\frac{\text{deg/sec}}{\text{deg/sec}}\right] \qquad (7)$$

Closed Loop

$$Y_{CL} = \frac{Q}{Q_C} = \frac{(G_E + Y_N)Y_C}{1 + (G_E Y_C)} \qquad \left[\frac{\text{deg/sec}}{\text{deg/sec}}\right] \qquad (8)$$

Error/Input Loop

$$Y_{IE} = \frac{Q_E}{Q_C} = \frac{1 - Y_N Y_C}{1 + G_E Y_C} \qquad \left[\frac{\text{deg/sec}}{\text{deg/sec}}\right] \qquad (9)$$

5. Numerical Values

The numerical transfer functions for the simplified fighter case are given in Table 1. These are given as s-domain factored polynomials, using the shorthand notation described in McRuer, et al, [1973].

High Freq. Gain: $K|_{s\to\infty} = K$

First Order: $(s + a) = (a)$

Second Order: $[s^2 + 2\zeta\omega s + \omega^2] = [\zeta, \omega]$

The Table 1 transfer functions do not include sensing, or sample-and-hold effects.

TABLE 1. NUMERICAL TRANSFER FUNCTIONS

Full System

$$Y_{OL}(s) = \frac{82.5\ (14.2)}{(.309)\ (8.33)\ (31.3)}$$

$$Y_{CL}(s) = \frac{82.5\ (14.2)}{[.817,\ 6.55]\ (29.2)}$$

$$Y_{IE}(s) = \frac{(.309)\ (8.33)\ (31.3)}{[.817,\ 6.55]\ (29.2)}$$

VEC Failed (Stab Alone)

$$Y_{OL}(s) = \frac{61.4}{(.309)\ (31.3)}$$

$$Y_{CL}(s) = \frac{61.4}{(2.44)\ (29.1)}$$

$$Y_{IE}(s) = \frac{(.309)\ (31.3)}{(2.44)\ (29.1)}$$

STAB Failed (VEC Alone)

$$Y_{OL}(s) = \frac{21.1}{(.309)\ (8.33)}$$

$$Y_{CL}(s) = \frac{21.1}{[.888,\ 4.86]}$$

$$Y_{IE}(s) = \frac{(.309)\ (8.33)}{[.888,\ 4.86]}$$

For the computing interval of $\Delta T_s = .05$ sec, the effective loop delay is about $\Delta T_s/2 = .025$ sec. This is concatenated with Y_{OL} as a first order Padé Approximant for the delay:

$$\exp(-T_s/2) = \frac{1 - (T_s/4)s}{1 + (T_s/4)s} \quad \cdots = \frac{1 - .0125s}{1 + .0125s} = \frac{-(80)}{(80)} \tag{11}$$

Parallel operation of two servos having different time-constants gives rise to a transfer function first order zero at 14.2 rad/sec, representing the net effect on the opened loop transfer function <u>without</u> ACN as shown below, with the computing delay added:

$$Y_{OL}(s) = \frac{Q}{Q_E} = \frac{82.5(14.2)}{(.309)(8.33)(31.3)} \left[\frac{-80}{80}\right] \underset{-M_q \quad 1/T_{STAB} \quad 1/T_{VEC}}{} e^{-T_s/2} \tag{11}$$

6. Simulation

The simulation of this one DOF aircraft Low Order Equivalent System (with the two first order servos) is very straightforward. The digital difference equations are as follows in the time domain (where n is the past sample and angular units are in radians):

<u>Aircraft</u>

$$\begin{aligned} \dot{Q}_{n+1} &= M_q Q_n - (M_{\delta VEC} VEC_n + M_{\delta STAB} STAB_n) \\ Q_{n+1} &= Q_n + \Delta T_S \dot{Q}_{n+1} \end{aligned} \tag{12a,b}$$

<u>STAB Servo</u>

$$\begin{aligned} \dot{STAB}_{n+1} &= 31.25\ (STABSUM_{n+1} - STAB_n) \\ STAB_{n+1} &= (STAB_n + \Delta T_S\ \dot{STAB}_{n+1}) \end{aligned} \tag{13a,b}$$

<u>VEC Servo</u>

$$\begin{aligned} \dot{VEC}_{n+1} &= 8.33(VECSUM_{n+1} - VEC_n) \\ VEC_{n+1} &= VEC_n + \Delta T_S\ \dot{VEC}_{n+1} \end{aligned} \tag{14a,b}$$

The computing interval is $\Delta T_s = .050$ sec or 20 Hz.

7. Validation of the Simulation

The frequency response Bode plot for the opened loop is given in Fig. 2a, along with the data points measured by the seven-frequency quasi-random forcing function. (See Section III-A for further details of this describing-function and measurement procedure.) The agreement is excellent, thus verifying the simulation of the controlled-element dynamics. Also shown in Fig. 2 are the closed-loop (Y_{CL}) and error/input (Y_{IE}), frequency responses. The agreement with the theoretical

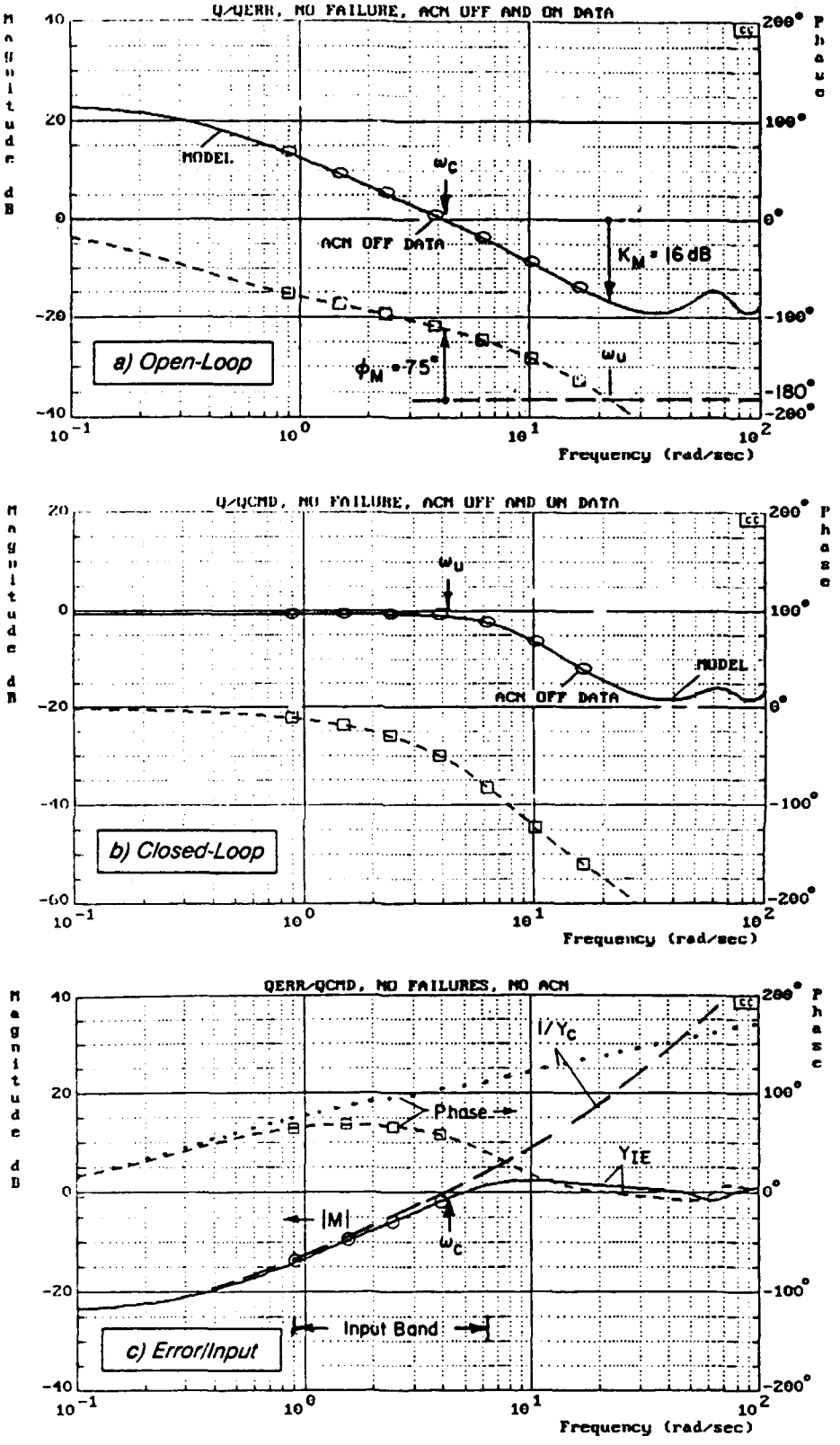


Figure 2. Validation of Simulation Model vs Measured Transfer Functions

digitized model is good throughout. Notice that over the lowest five frequencies, the magnitude curve of Y_{IE} is nearly the inverse of Y_{OL}, but the phase curves do not correspond to each other. This observation will be used later.

The importance of early verification of the controlled element by independent frequency domain measurements was underscored, here, because several subtle coding errors were revealed by mismatch of the computed vs. measured frequency responses. It is also essential that the delay corrected form (Z-transform) be used to allow for high-frequency phase distortion due to the computing delay artifact.

C. IMPLICATIONS FOR A FAILED CONTROL

For this LOES aircraft, pitch stability is neutral ($C_{M_\alpha} \doteq 0$), so a failed STAB or VEC drastically reduces the effective gain of the controlled element. This, in turn, reduces the opened-loop gain magnitude ratio and increases the Y_{IE} magnitude. Here: a) the rms error for a sum-of-sines input is the root-sum-squared of the discrete input frequencies; b) a Bode plot is a logarithmic representation of the error magnification so the highest few frequencies dominate the rms error; and c) the inputs (here) have constant or "white" amplitude. Thus, the input frequencies at or beyond the crossover frequency, ω_c, will dominate the rms error in such cases.

Keep this observation in mind later, while interpreting the effects of STAB or VEC failure effects with the ACN "off" and "on."

D. IMPLICATIONS FOR PLANT INVERSION

An interesting property of a tightly closed control loop, one in which the error is much less than the input over a bandwidth mostly below the unity gain crossover frequency, is that the error/input transfer function (YIE) is a close approximation of the inverse of the opened-loop transfer function, $1/Y_C$.

1. Let $Y_{OL} = G_E(s)Y_C(s)$, where G_E is the controller, suitably equalized to yield a "Crossover Model" type of Y_{OL} [McRuer, et al, 1973]. This implies:

   i. gain >> 1 at low frequencies, below ω_c
   ii. gain << 1 at high frequencies, above ω_c
   iii. adequate phase margin, Θ_M, near ω_c for ζ_{CL} = .3 to .8 is 30° to 80°
   iv. adequate gain margin, K_M, near ω_u of 6 - 20 dB

   Here, very conservative stability margins of K_M = 16 dB and Θ_M = 75° were used, to allow for possible NC destabilizing effects.

2. The closed loop error transfer function (NC = OFF) is given by:

$$Y_{IE}(s) = \frac{1}{1+Y_{OL}(s)} \tag{15}$$

For inputs below ω_c; where $Y_{OL} >> 1$ per i above:

$$Y_{IE}|_{\omega<\omega_c} \approx \frac{1}{Y_{OL}} \qquad \omega \ll \omega_C \tag{16}$$

But for inputs well above ω_c, $Y_{OL} << 1$ per ii above:

$$Y_E|_{\omega>\omega_c} \approx 1 \quad (M \rightarrow 0\text{dB, phase} \rightarrow 0^\circ) \qquad \omega \gg \omega_C$$

For the baseline case, refer to the similarities and differences between $1/Y_{OL}$ and Y_{IE} in Fig. 2c.

Keep these fundamental relationships in mind, as they form the basis, here, for the neurocontroller training scheme and its limitations, as well.

III. IDENTIFYING CONTROL PROCESS DYNAMICS

A. FORCING FUNCTIONS

1. Rationale and Selection

In developing and analyzing neural net controllers, the use of appropriate forcing functions (commands and disturbances) is important, because neural nets can "learn" specific inputs and optimize the control actions for them. Among the candidates for the Phase I demonstration of fighter control were: various discrete maneuvers (steps, doublets, [1-cosine] functions, etc.), and quasi-random inputs. The detailed characteristics of these test functions are important, since inadequately structured inputs can often obscure adverse system qualities.

Quasi-random commands to the pitch rate were selected for the following reasons:

i) They are typical of the pilot control task, in maintaining line-of-sight attitude against target aircraft disturbances.

ii) They can be designed to cover a wide range of the command-space of amplitude, rate, and frequency.

iii) They have a much broader utility for identification of the adaptive neurocontroller properties across the range of command space than simpler, more repetitious forcing functions.

Although other forcing functions will be recommended for use in Phase II, quasi-random pitch rate commands were used for the Phase I experiments.

A special type of quasi-random signal, termed a "sum-of-sines" [SOS], was used in the present study. In this type of signal, several sinusoids of different frequencies are summed. The selection of frequencies is critical and something of an art, because one must avoid any low and simple harmonic ratios, otherwise repeated patterns will occur. The number of sinusoids must be limited to 5-10 waves in order to concentrate the input power for better signal/noise ratio in measured responses of nonlinear or noisy systems. Near logarithmic spacing in frequency is desirable to best exhibit the frequency response on a Bode plot with the fewest frequencies. Finally, an integer number of cycles per run length is needed to provide accurate Fourier-integral coefficients.

STI has over 30 years experience in developing and applying this sum-of-sines approach for use as a system identification tool [Allen and Jex, 1972]. Although other test signals were initially applied to this problem, it became apparent that the SOS would likely be more productive in revealing non-obvious system characteristics.

2. Specific Inputs

We used two sets of inputs: a 7-sine wideband case for verifying the controlled element simulations, and a 5-sine baseline set with the highest frequency near the crossover region, for the reasons discussed earlier. By using all sinusoids of alternating polarity, the compound waveform

exhibits several desirable properties. First, it starts and ends at zero; second, the beginning and ending portions have low slopes; and third, the overall waveform is inherently skew-symmetric. The low initial and terminal slopes condition the waveform for better accuracy of the Fourier integrals and the skew-symmetry guarantees a symmetric amplitude distribution. However, the rates or integrals of the sines are all cosines, which produce a bilaterally symmetric waveform over the epoch and a non-symmetric amplitude histogram. For convenience, all sinusoids had the same amplitude, adjusted to achieve the same RMS level. This amounts to a sort of "pink" noise, because the constant power sines are divided by a logarithmically increasing bin width. Because the total power in the waveform is concentrated at several precise frequencies, the signal/noise ratio for noisy systems is excellent.

The resulting forcing functions are designated as follows:

"SS-5" 5 Sum-of-Sines - frequencies from ω = .90 to 6.28 r/s (.14 to 1.00 Hz), RMS = 1.87 deg/sec

"SS-7" 7 Sum-of-Sines - frequencies from ω = .90 to 16.2 r/s (.14 to 2.57 Hz), RMS = 1.87 deg/sec

Table 2 gives the parameters for each case.

Table 2 Sum-of-Sines Input Parameters

Sine No.	1	2	3	4	5	6	7
sign	+	-	+	-	+	-	+
Amplitude: "SS7"	1.0	1.0	1.0	1.0	1.0	1.0	1.0
"SS5"	1.18	1.18	1.18	1.18	1.18	0	0
No. of Cycles in Epoch (T)	3	5	8	13	21	34	55
Frequency - Hz (T = 21.0 sec)	.143	.238	.387	.619	1.000	1.571	2.571
Frequency - rad/sec	.89	1.50	2.39	3.89	6.28	9.87	16.16

Statistics	5 Sum-of-Sines	7 Sum-of-Sines
Epoch Length	21	21
Mean	0	0
Std. Deviation	1.87	1.87
Skewness	0.0	0.0
Kurtosis	-.77	-.94

The run length epoch should be long enough for at least three cycles of the lowest frequency; here, it is T = 21 seconds. Notice that the number of cycles per run is integer, i.e., 3, 5, 8, 13, 21, 33, and 54 for SS-7. These are the seven Fibonacci Number Series, from 3 to 54 in which each value is the sum of the preceding two values. These have no simple harmonic ratios, but the ratio

of succeeding frequencies is always near 1.6 and approaches 1.618 eventually, giving even spacing on a logarithmic Bode plot.

The amplitudes of each sine in the 5 Sum-of-Sines is increased by a factor of 1.18 = $\sqrt{7/5}$ over those in the 7 Sum-of-Sines to equalize their rms values to be equivalent to $\sigma_5 = \sigma_7 = 1.87$ degrees/sec of input.

The sum-of-sines generator for the simulation is in two subroutines in Appendix B: "ffSumofSins5" and "ffSumofSins7." A running rms computation with a 3 second window is also applied, as part of post-run data processing.

3. Examples

Time histories of the 5-SOS and 7-SOS inputs are given in Figures 3 and 4. Shown in each case are: the time trace, the running rms, the amplitude histogram, and the power spectral density. The following points are noted:

- The waveform has a gentle start and finish, as intended.
- The distribution of large and small peaks is quite uniform, as confirmed by the 3-second running rms signal.
- The amplitude histograms are roughly monotonic and are slightly more leptokurtic than a pure gaussian distribution.
- The PSD are pure line spectra, nearly evenly distributed in the (logarithmic) Bode plot, as intended.

Thus, these sum-of-sines forcing functions appear nearly ideal for use in neurocontroller training and validation.

B. MEASURES OF NEURAL NET ACTIVITY AND EFFECTS

The adaptive clustering network (ACN) adds a new processing element (PE) whenever a point in input space occurs which does not fall within the receptive fields of any existing PE. This is referred to as "bottom-up instantiation." In addition, the ACN, as presently implemented, is capable of PE instantiation based on the feedback error signal. This is referred to as "top-down instantiation" and is a feature of STR's proprietary ACN architecture which enhances its ability to rapidly adapt to mapping nonlinear regions of the system's input/output space. By monitoring the growth rate of new cells and the source (bottom up or top-down) of the instantiation request, we can determine the appropriateness of the ACN structure to the I/O space and its rate or learning at any instant in the simulation epoch. The "characteristic radius" of the gaussian shaped neuron-influence "transfer functions" is conventionally specified as σ, the inflection point of the gaussian function. Sufficient neurons are added to assure that at least one or more overlapping radii cover the locus of input states traversed by the training process.

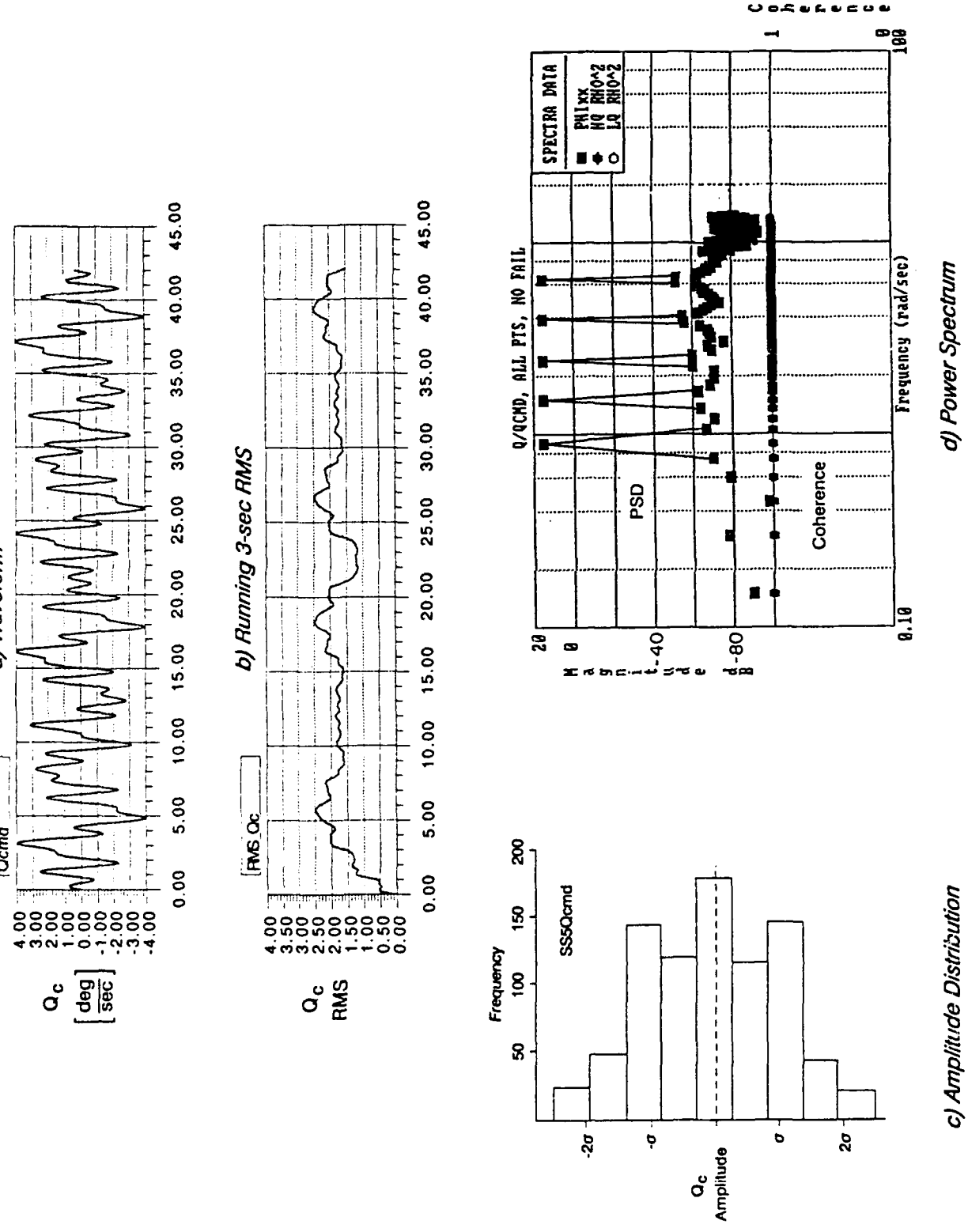


Figure 3. Properties of the Five Sum-of-Sines Forcing Functions

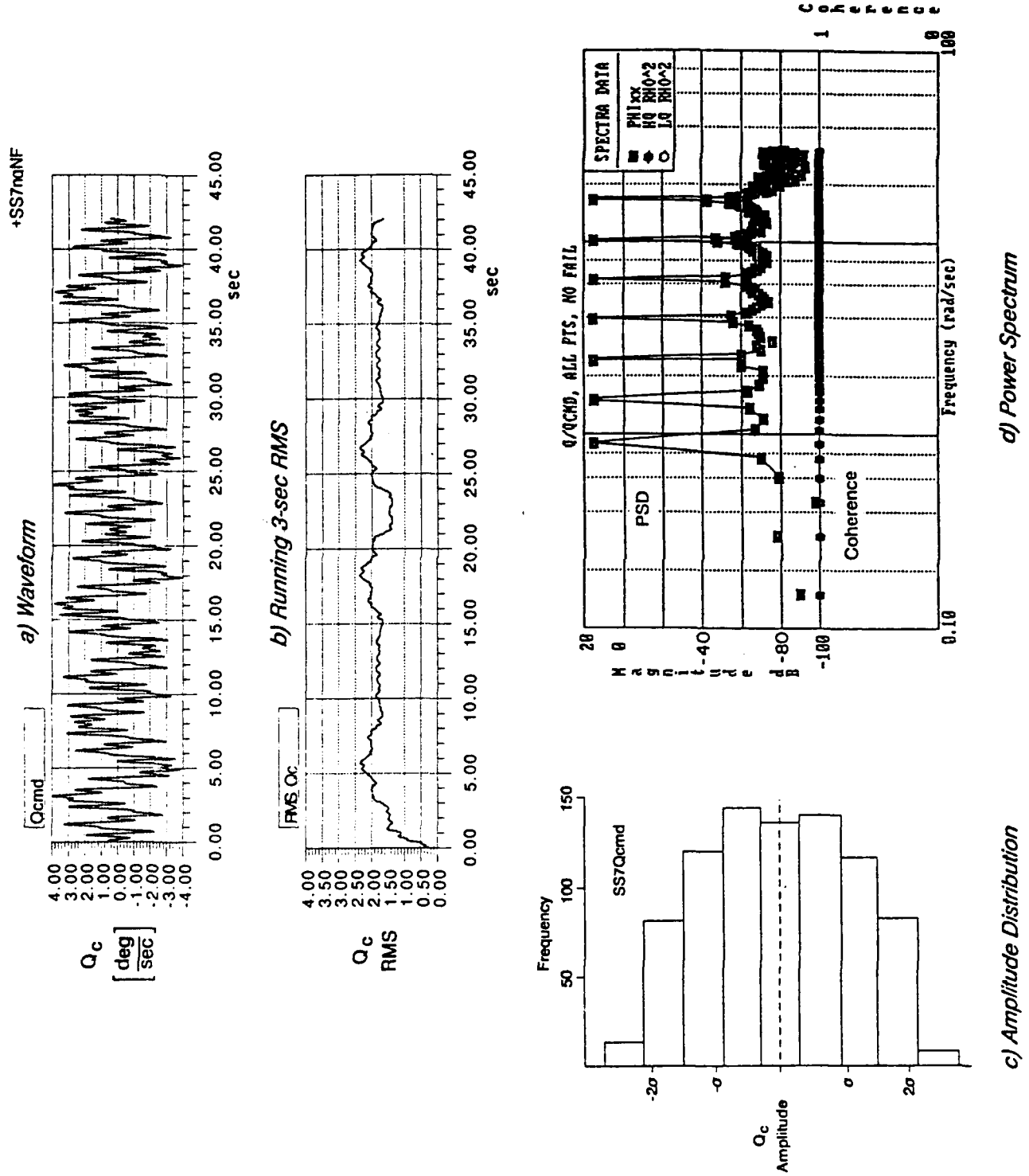


Figure 4. Properties of the Seven Sum-of-Sines Forcing Functions

While the neuron radius σ is adjusted as a parameter, here, the ACN algorithm determines the centroid loci. Therefore, a plot of their locations in the NN input region is a useful measure of the ACN distribution and will be shown later.

Many of the neural net effects are subtle and nonlinear and can only be appreciated by close comparison of the waveforms of the process during training or following a failure of some type. Time traces of the relevant pitch rate command, response and error signals and the running rms error will be used for a side-by-side comparison of neural net "off" and "on" effects. Corresponding side-by-side time histories of the control signals from the feedback-error controller and the neurocontroller, are also given for each case.

For this report we have restricted the time histories to a 42 second run, which includes two of the basic 21-second input epochs. This permits accurate frequency domain describing functions to be obtained for the early and late portions of a long training run, or for the pre- and post- failure portions of a failure run (the failure always occurred at 21+ seconds).

Finally, the various frequency response describing functions were computed by STI's proprietary FREquency Domain Analysis (FREDA) program, using the relationships defined in Figure 2 and Equations 5 - 9.

IV. NEURAL-NET CONTROLLER

A. CONTROL LOOP ARCHITECTURE

Application of ANN based controllers, or for that matter, any adaptive element to problems for which conventional control solutions already exist requires that careful attention be paid to the placement of the adaptive element in the control loop architecture. Our proposed approach was a doubly cautious one in which we would: (1) apply a neural net only in a position where it could select from within a predetermined set of parameters already determined to be appropriate for stable system operation and (2) use a neural net architecture already of proven value in a feedback control context.

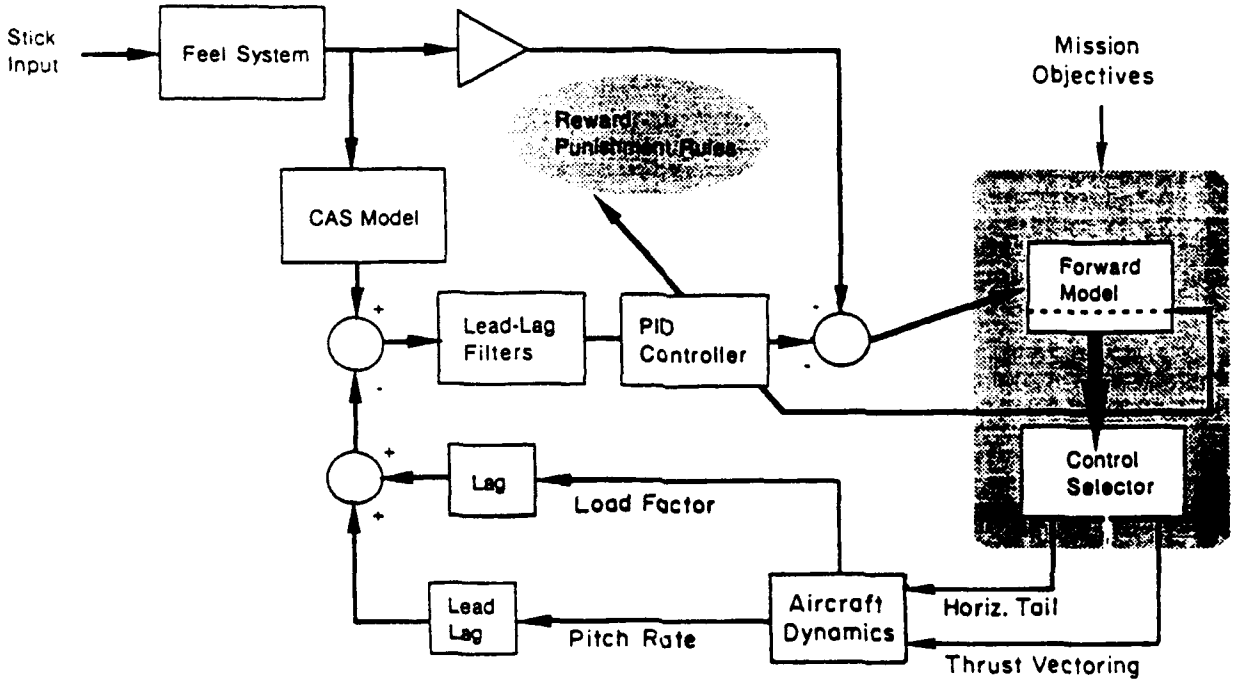


Figure 5. Proposed Flight Controller with ANN-Based "Intelligent Configuration Management System " (ICMS)

Figure 5 shows a typical aircraft control system augmented by the forward model. In this architectural concept, we proposed to place the primary adaptive element in the control loop as a forward model. This construction provided for a forward model of the plant whose function was to adapt in response to critical cues provided by external performance criteria. Output from this model would then

influence the intrinsic response of the plant in two ways. First, parametric adjustments would be provided to a PID controller element on order to compensate the system for altered frequency domain requirements. Second, state characteristics of the forward model would feed forward into a preprogrammed nonlinear selector module and serve as flight mode-changer, allowing intrinsic control system coefficients to be altered by learned patterns of control system activity. In this model framework, it was planned to have *the ANN function in a forward model loop and feed back into a conventional controller.* It was felt that this architecture would be advantageous in re-identification of the system following component damage and would allow for maximum reliance on existing aircraft states while allowing the adaptive element to develop a modified *state image* and simultaneously minimize the likelihood of margin exceedances. In addition, the forward model architecture could be easily adapted to the reduced-order form used by STI to explore experimental models. This modeling approach essentially represents the method used by Jordan [1989] and may be viewed in more conventional block diagram representation as shown in Figure 6 below.

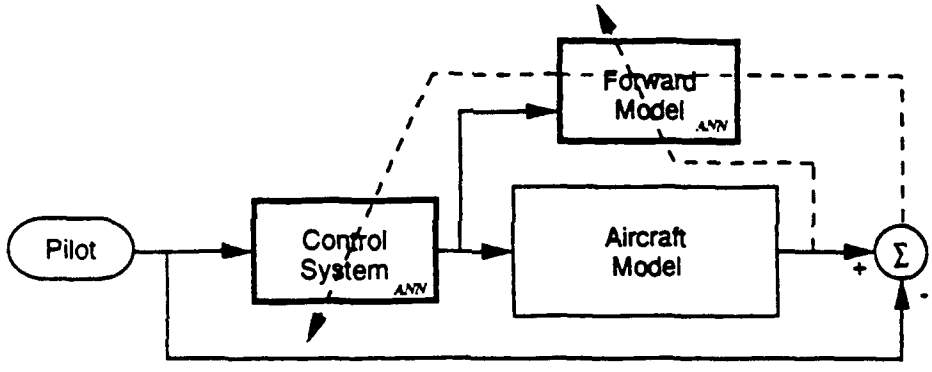


Figure 6. Forward Modeling ANN Loop Structure

Although this approach had certain advantages, it also had some non-obvious disadvantages. The most obvious advantages included inherently definable state space representations of all flight modes, tangible limits on the control authority developed by the adaptive element, and the use of conventional gain scheduling technology. Disadvantages, however, included the use of an adaptive element based on *backpropagation* which could be inherently slow to adapt, especially at conventional FCS frame rates; the requirement for external sensors and AI input to manage the mission-related sensitivity of the system and the additional development required for appropriate externalized reward-punishment without guarantee of their generality. The results of backpropagation based forward modeling experiments presented in the literature suggested that rapid convergence is difficult to achieve. Our own initial experiments with

backprop-based forward modeling reinforced this conclusion, since several of these cases failed to converge in preliminary experiments.

Other approaches which were considered included other controller configurations based on standard backpropagation. In backpropagation, the partial derivatives of an error criterion are used to adjust the weights along an error gradient in order to minimize the error function. Although backpropagation is considered to be a computationally efficient scheme for performance based ANN learning [Narendra, 1990], we found it to be poorly suited to the aircraft problem in this context for the following reasons:

- Backpropagation networks are trained using a form of gradient descent, resulting in long training times and the possibility of becoming trapped in a suboptimal state called a "local minima." Our preliminary experiments using backpropagation in the feedback error configuration showed an unacceptable rate of convergence on the error criterion which we had established for performance of the controller.

- The oscillatory changes undergone by the feedback error controller with a feedback-error element based on backpropagation detracted from the stability of the aircraft, especially during early stages of learning. In the context of damage management, it was felt that this was inappropriate behavior during this critical phase of mission management.

Even though backpropagation represents one of the most common implementations of ANN architecture in specialized hardware, we felt that its performance was sufficiently adverse in this context as to warrant development of a more customized approach. After careful review on the performance characteristics of various network architectures, we determined that certain types of networks had adaptation qualities considerably better suited to real-time flight control than the standard backpropagation configuration. These qualities fell into the following performance areas:

- Mapping capability
- Convergence
- Distribution of adaptation rates during training
- Noise characteristics

Since the aircraft problem, as posed, required the model to address continuous control issues as well as issues relating to parameter identification and discrete state classification, we focused our review of alternative architectures on those with capabilities of rapid learning with useable, if not accurate, transitional state output. In our review of the literature dealing with neural nets in control applications, we found a variety of architectures for incorporating ANNs into the control scheme. One of these is discussed in Kawato's treatment of robotic arm trajectories [Kawato, 1990], specifically, the use of the ANN to learn the inverse dynamics of the system. A typical Kawato feedback error configuration is shown in Figure 7.

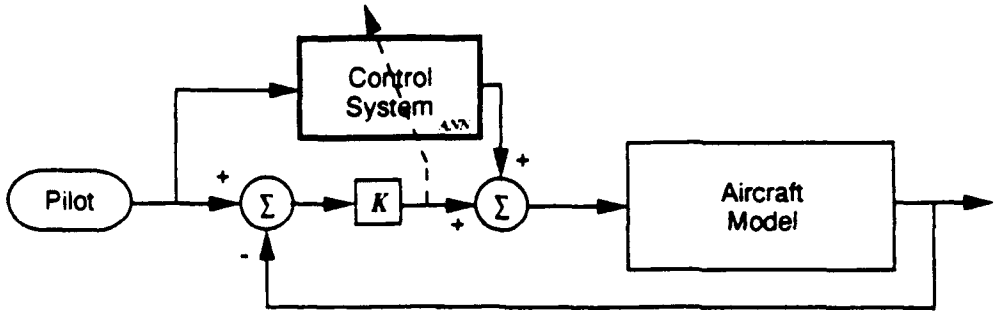


Figure 7. Kawato Type-C ANN Loop Structure

Note that this approach requires the existence of a conventional controller to "bootstrap" the system.

In the Kawato Type C architecture, [Kawato, 1990], it is assumed that in a tightly closed control loop, the signals from the controller approach $1/Y_{OL}$ and can, therefore, substitute for the plant inverse. This approximation is valid for the demonstration case, however, when the command input approaches the crossover frequency of 4.3 rad/sec, the feedback error approaches 1, and the approximation is no longer valid.

Kawato, et al, in an earlier report, [Kawato, et al, 1987], recognized that when a tightly closed loop was presented that the inverse mapping could be successfully approximated by this method. Since the example system used in the present report appeared to have the desired properties of low gain and relatively tight loop closure, we felt that the Kawato Type C loop architecture was a good choice for the experiments reported herein.

B. THE ADAPTIVELY CLUSTERED NEURON (ACN) MODEL

The ACN model is hierarchical, using a simple fan-out input layer and two heterogeneous computing layers. The first is a self-organizing hidden layer comprised of nodes, each of which has the ability to adjust its receptive field based on a utility function. Processing elements which conform to these operating characteristics are sometimes called *Radial Basis Function* (RBFs) [Musavi, et al, 1991]. In addition, this hidden layer contains a variable number of RBF processing elements (PEs) and under unsupervised learning, responds to a rule-set which allows for the addition of new PEs to the layer [Hirose, 1991]. Figure 8 schematically illustrates the structure of the ACN as implemented in the present studies. The input and output layers conform to Grossberg *Instar* and *Outstar* configurations, respectively. [Grossberg, 1974]. The baseline ACN configuration utilizes a two-element input layer for pitch-rate command (Q_C) and its computed derivative ($\dot{Q}_C$). This layer is fully interconnected with an RBFN layer of variable size which can be augmented whenever the input space is not appropriately mapped and/or when certain other rule-based conditions are met. These augmentation rules are based on either "bottom-up," data-driven

control signals or "top-down" error signals [Holdaway, 1989]. This self-organizing hidden layer is followed by a supervised learning layer capable of rapid, "top-down," performance based learning. ACN network architecture conforms to the same overall design objectives as Hecht-Nielson's counterpropagation model [Hecht-Nielson, 1987]. The table at the bottom of Figure 8 shows certain network parameters which can be experimentally varied in the simulation code listed in Appendix B.

The ACN's hidden layer uses unsupervised learning methods to adaptively cluster the inputs. Using the principles described by Holdaway [Holdaway, 1989], the clustering algorithm is driven by an error signal (top down) and by the data input stream (bottom up). This strategy is thought to increase learning speed in addition to providing statistically optimized preprocessing of input signals. The output layer of the ACN may be constructed in several ways, depending upon design constraints. Among the output layer structures which have been applied to RBFN models are:

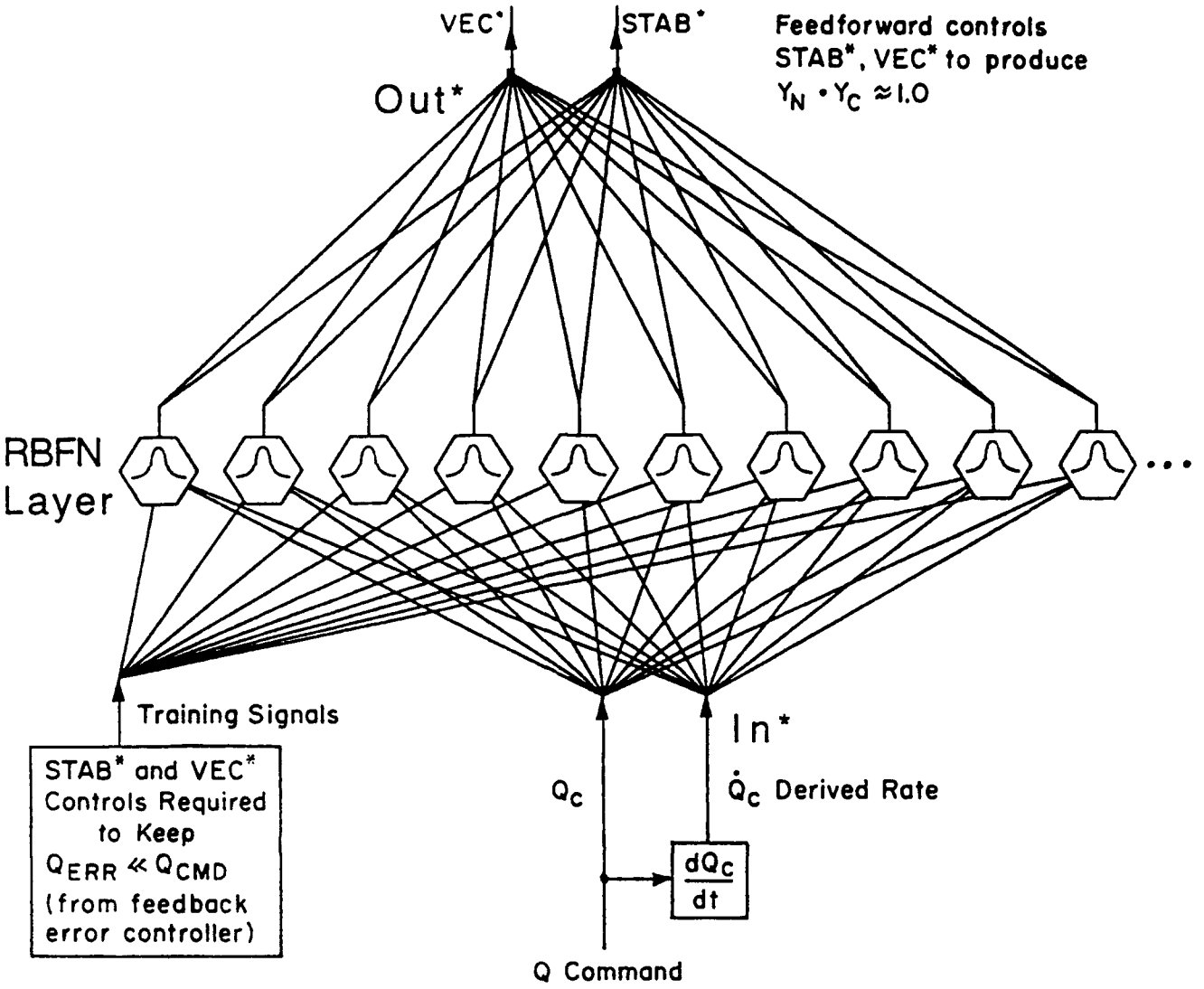
- Grossberg outstar
- Backpropagation
- Barycentric
- Widrow-Hoff (Performance Learning)

Each of these architectures exhibits specific advantages and disadvantages for various types of dynamics. For the present example, we have used the Widrow-Hoff structure, implementing the output layer as a "vector follower," with learning gated by pre-synaptic activation levels. STR also considered implementing a Barycentric model to distribute RBF layer activations. It was initially theorized that the non-linear Barycentric PEs would provide superior interpolative mapping of hidden layer neurons. In order to use the Barycentric model, it is necessary for the ACN to calculate a full variance - covariance matrix pseudo-inverse for use by the Barycentric PEs. This would necessitate a multi-rate implementation of the ACN, which could exhibit loss of stability due to inter-sample learning. For this reason, the Widrow-Hoff model was selected. A recurrent version of the ACN was also tried. It gave no clear performance advantage. We must remember, however, that in these experiments, the ACN was presented with input-output mappings with a persistent phase error. The apparent failure of this system to completely map the plant inverse, therefore, cannot be wholly attributed to either the loop architecture or the ACN structure.

In the Widrow-Hoff performance learning model, an error signal is computed for each iteration and applied through a weight adjustment algorithm which attempts to reduce the error on the next iteration. To implement Widrow-Hoff learning, a weight matrix [W] is maintained in input/output space for each set of input (x) and output (y) signals. For each iteration of the system simulation, the error vector $\underline{e}$ represents the difference between the actual and desired ouputs, so:

$$\underline{e} = \underline{y} - [W]\underline{x} \tag{17}$$

The weight matrix, [W] is updated using the algorithm $\Delta[W] = k\,\underline{e}\,\underline{x}^T$, where k is a small, positive gain coefficient used to modulate learning rate and control error overshoot during early phases of learning.



RBFN Inputs and Outputs Parameters (Selectable)	Default	Selected
Maximum Hidden Layer Size	200	
Capture Radius (θ_N/RMS signal)	5.0	1.0
Initial Centroid Weights	0.5	
Initial Radius Weights	0.9	
Alpha (base threshold for Gaussian capture region)	[computed]	
Activation Rate (Widrow-Hoff)	.01 * dt	
Aging Rate	0.1 * dt	

Figure 8. Baseline Configuration of RBFN Inputs and Outputs

Alternatively, this can be expressed as:

$$W_{ij}' = W_{ij} \cdot \alpha \cdot x_j \cdot (d_i - x_i)$$

or, as in the present implementation:

$$W_i^{NEW} = W_i^{OLD} + (\alpha \cdot y_i \cdot e') \tag{18}$$

where e′ represents the returning error signal from the previous iteration, i.e., k_i (Q_c^{OLD} - Q^{OLD}).

Widrow-Hoff learning accomplishes a gradient-descent approach to output layer weight optimization. Given stationary dynamics, it is guaranteed to converge on the unique set of weights [W] which will yield the minimum RMS error to a steady state signal.

In the present case, where the dynamics are time-variant, this convergence is not guaranteed, however, the gradient-descent properties of the Widrow-Hoff algorithm were considered a good justification for adopting this strategy for regulating output layer adaptation.

The simulation program used in the present study is listed in [Walters and Smith, 1991]. This program represents a customized adaptation of several proprietary ANN methods developed by STR for control applications. It should be emphasized that this program is a specially tailored example for the preliminary studies reported in this paper, and does not reflect many of the advanced features and recent innovations in STR's neurocontroller development software. For example, strategies for noise suppression, cell annihilation, capture region shrinkage, and more selective cell commitment are presently under study.

V. RESULTS

A. VERIFICATION OF NN OPERATION

The effects of neurocontroller actions are inherently time varying (as it "learns"), nonlinear (control actions are not necessarily proportional to inputs or errors), and often subtle (affecting details of waveforms). Therefore, we first compare the relevant time histories with NC = "OFF" vs "ON." Next, the various measures of performance and NN functioning are compared. Finally, we analyzed selected signals by frequency-domain describing functions to validate certain hypotheses and to interpret what was happening in the neurocontroller block.

1. Effect of "Baseline" Neurocontroller, With No Failures:

The "baseline" NC configuration adopted was the basic Kawato Type C setup (inputs of command state and derived rate; trained in the error feedback control commands), using quasi-random inputs.

Baseline Case:

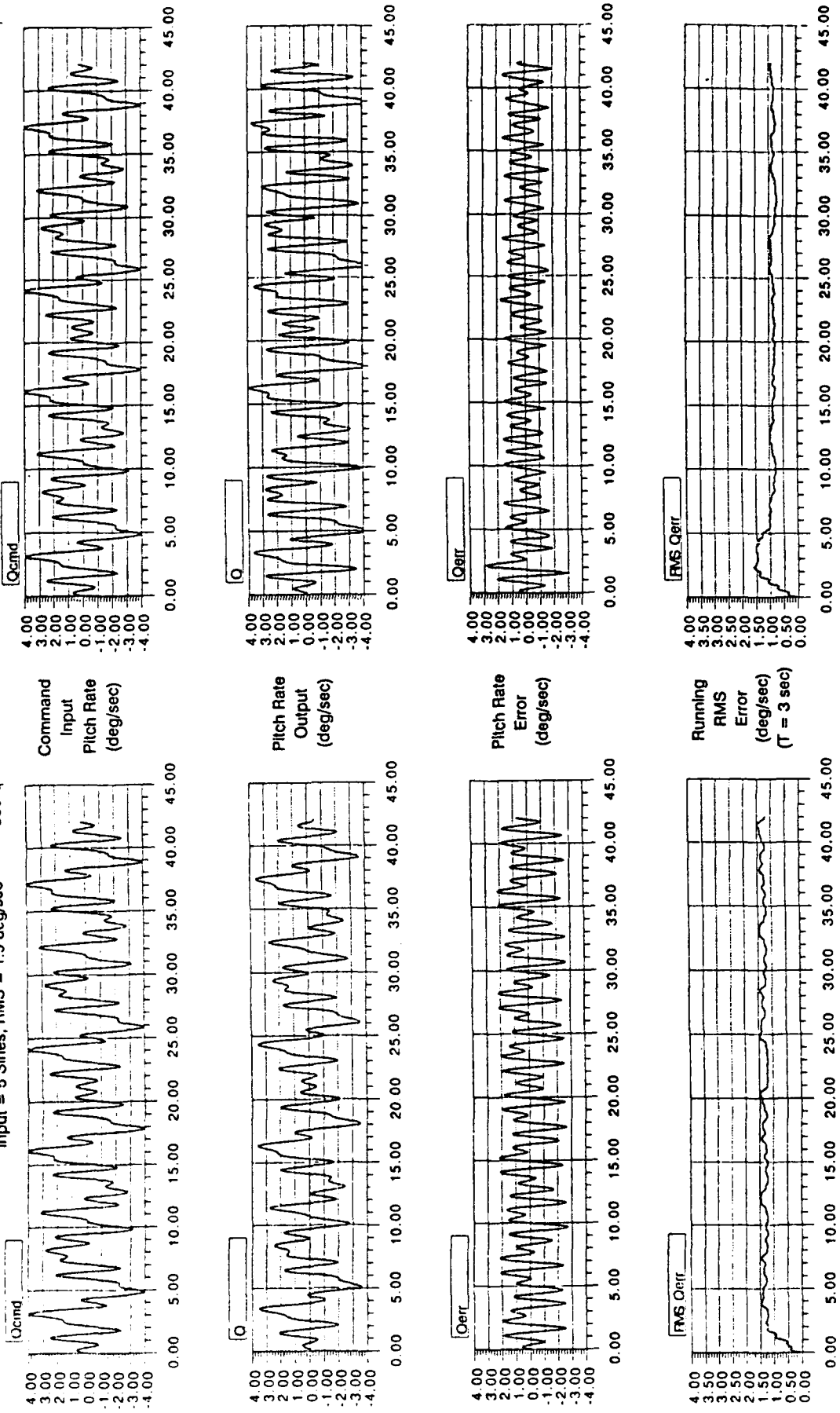
Neuron effective radius:	$r = 1.0\sigma_{Q_c}$, $1.0\sigma_{Q_c}$
State inputs to NC:	Q_{CMD}, $\dot{Q}_{CMD}$
Training criterion to NC:	$C_E = STAB_{CMD}$, VEC_{CMD}
	from the Q_E feedback error controller
Forcing function (5 Sum-of-Sines):	ω_i = .9 - 6.3 rad/sec (.14 - 1.0 Hz)
	σ_{QC} = 1.87 deg
Run length (2 input epochs):	42 sec

The five component forcing function permitted easy plotting of NC properties vs Q_C, $\dot{Q}_C$, and the lower bandwidth was more typical of pilot commands as well as mostly below crossover frequency.

First, carefully examine the time histories in Figures 9 and 10, where NC OFF is on the left and NC ON is at the right. Corresponding signals in each plot are to the same scales to facilitate comparison. From the pitch rates of Figure 9-a and 9-b, the following points are noted (signal definitions are given in Fig. 1):

NC Finding #:

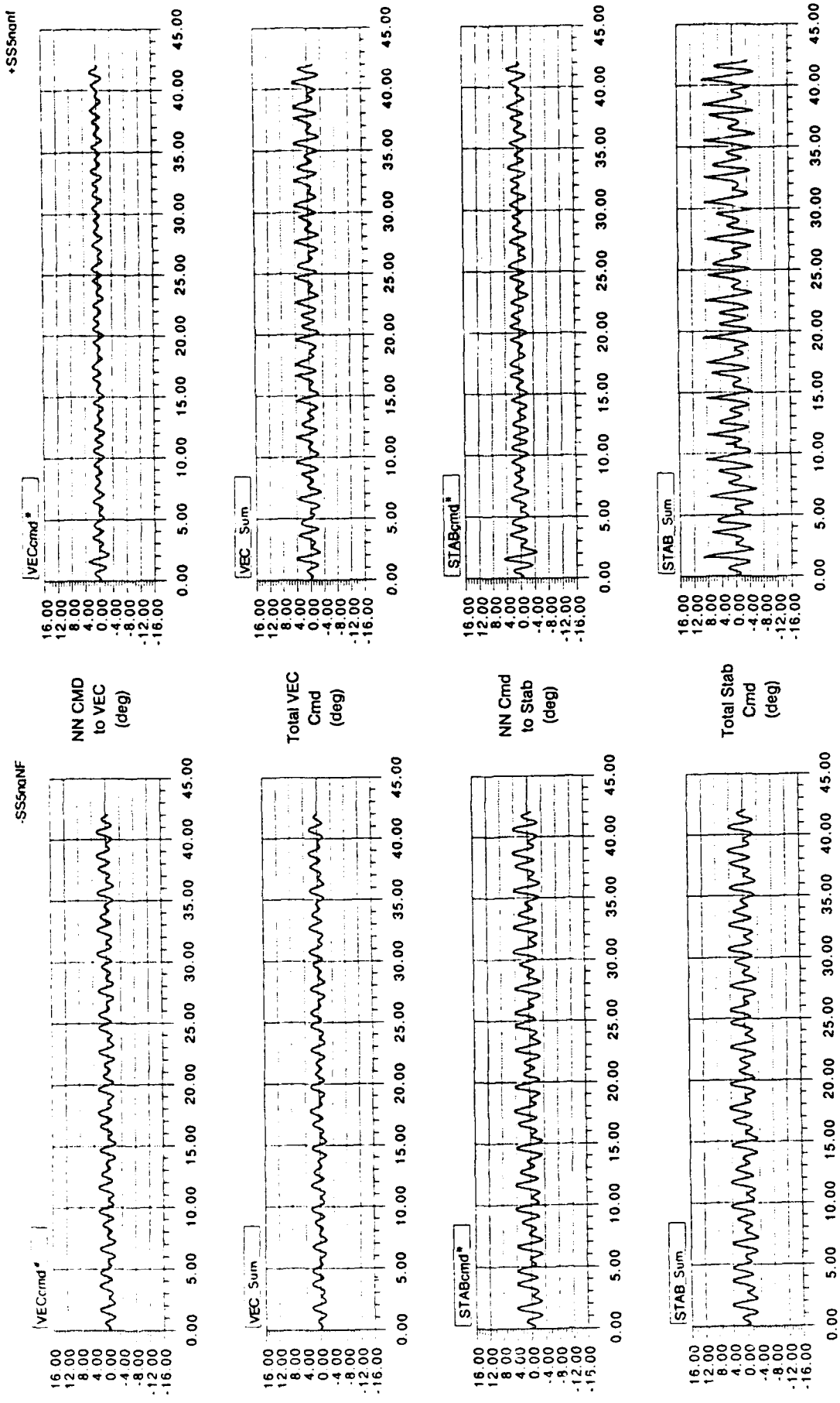
i. With NC OFF, the output Q appears to follow the command Q_C waveform fairly well. However, the tracking error Q_E is appreciable, as shown by the 3-second running RMS Q_E below. With NC OFF, the RMS error is about 70 percent of the command (1.3 / 1.87).



a) NN = "OFF" (feedback error correction only)

b) NN = "ON" (feedback error + NN feedforward)

Figure 9. Effects of Neural Net Operation on Pitch Rate Signals; Baseline Case, No Failures



b) NN = "ON" (feedback error + NN feedforward)

a) NN = "OFF" (feedback error correction only)

Figure 10. Effects of Neural Net Operation on Control Signals; Baseline Case, No Failures

ii. With NC = ON, the NC seems to very rapidly adapt, as evidenced by the improvements in the RMS error trace. Most of the adaptation seems almost complete by 10 seconds, as is demonstrated in later plots.

iii. The RMS error/input ratio is further reduced by NC = ON to about 46 percent (.86 / 1.87 = .70). This is not as much as might be expected if the NC were successful in completely inverting the plant as explained in Section II - D, but is clear evidence of successful NC Action in improving the control effectiveness.

iv. The error signal is dominated by the highest pair of input frequencies (.62 and 1.0 Hz). They are at and slightly beyond the crossover frequency, (.67 Hz or 4.2 rad/sec), so are not followed well. Nevertheless, if the plant were accurately inverted by the NC feedforward loop, these errors would be reduced, as well. So, this is evidence that the NC is not truly inverting the plant at higher frequencies.

Some key control signals are compared in Fig. 10: the $( )^{\bullet}$ signals are the control commands to STAB or VEC from the NC block. The SUM( ) signals are the total control commands from both the linear feedback error controller $( )_{CMD}$ and NC controller $( )^{\bullet}$. (The $( )_{CMD}$ signals are not shown because they are proportional to the Q_E signals in Fig. 9). The following points are concluded from Fig. 10:

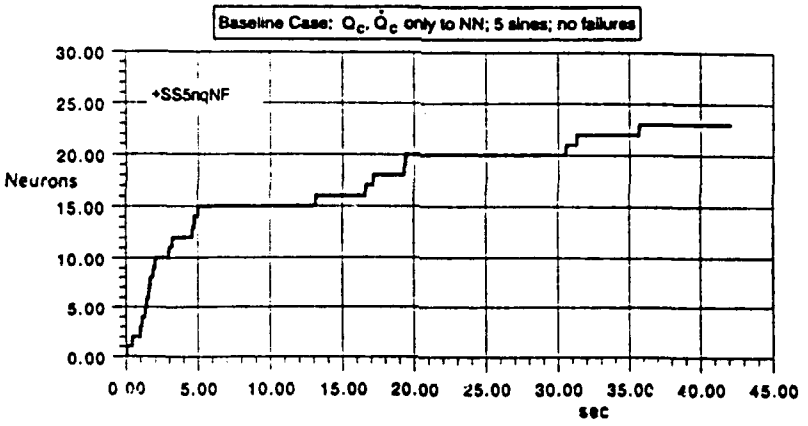
v. Because the Q_E is less with NC = ON, the $( )_{CMD}$ signals are reduced, as expected.

vi. The total control commands are larger with NC = ON, because the NC signals $( )^{\bullet}$ become larger as the NC attempts to "learn" the plant inverse. This is evidence of effectively increased loop gain.

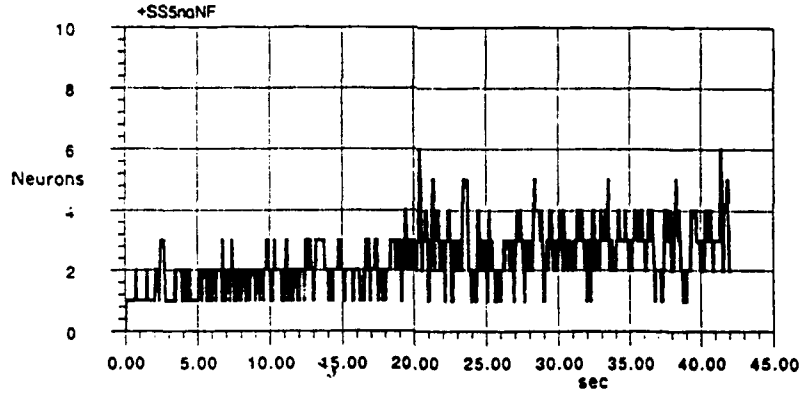
vii. Although there is some evidence of higher frequencies in the NC control output (especially for other cases not shown here), it is dominated by the forcing function frequencies. This observation justifies describing function analysis of the NC control outputs.

viii. Some (+ vs -) asymmetry is seen in the control signals. This reflects the earlier observation (Section II), that the error signal is composed of mostly cosine waves, which have an unsymmetric amplitude distribution.

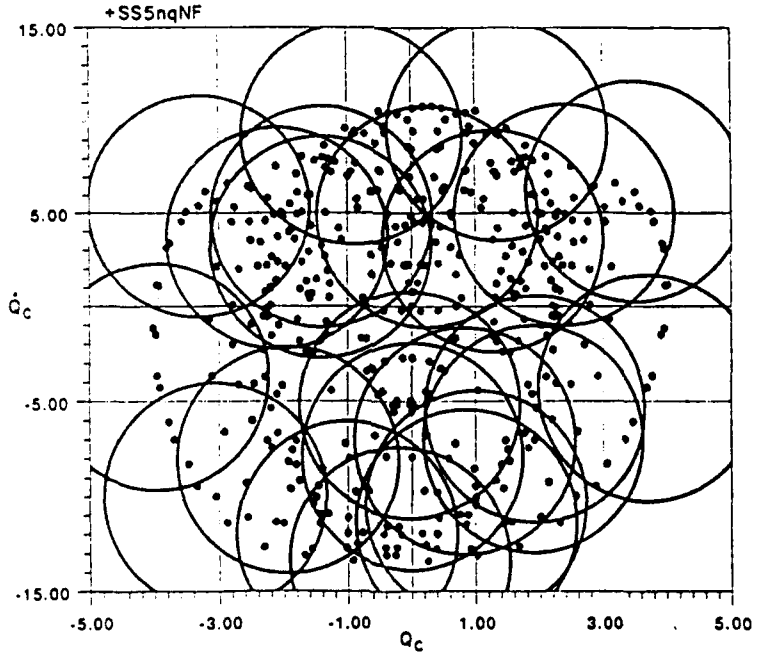
Further validation of the NC's action (to be discussed later) is that the $VEC^{\bullet}$ and $STAB^{\bullet}$ for NC = OFF (not fed to the servos) are exactly proportional to the VEC_{SUM} and $STAB_{SUM}$ signals. That is, the NC has been trained to recall the VEC and STAB control commands learned from the linear control loop's corrective commands. Some measures of the neurocontroller activity for this baseline case are given in Figure 11. At the top are the cumulative number of neurons added by the Adaptive Clustering Network algorithm (see Section IV). As surmised above, most of the adaptation (of the total 23 neurons) is done within the first 6 seconds, because the command state and rate space is quickly covered by the quasi-random forcing function.



a) Cumulative RBF Cells



b) Concurrently Active Cells



c) Centroids of ACN Cells After 21 sec

Figure 11. Neural Net Activity Measures; Baseline Case

The neuron coverage is illustrated in Fig. 11-c, showing each point in the trajectory of the forcing function in the Q_c, $\dot{Q}_C$ phase-plane as a dot for each pair of concurrent Q_c and $\dot{Q}_C$. The plots are scaled such that the rms Q_c and $\dot{Q}_C$ have the same plot distance. The ACN neuron's effective action zones are superimposed as circles of the effective radius $R_N = 1.0 \times \sigma_Q = \sigma_{\dot{Q}}$. The conclusions drawn from Figure 11 are as follows:

ix. From Figure 11-c the sampled trajectory points of the quasi-random 5-SOS forcing function states fairly uniformly cover the Q_C, $\dot{Q}_C$ state space, leading to a correspondingly fairly uniform distribution of ACN-clustered neurons. It is symmetric in Q_C (which is all-sines), but is asymmetric in $\dot{Q}_C$ (which is mostly all cosines). Some cosine waves should be added to the forcing functions to produce a more uniform result.

x. The 1-unit normalized neuron radius (ratio of effective RBF radius to each state's RMS signal level) provides almost complete coverage of the command state region, with sufficient overlap for 1 to 5 neurons to influence each control action.

xi. To cover the NC-input state space (here, Q_C and $\dot{Q}_C$ only), takes about 21 neurons of a 1 σ_Q effective radius and is 70% trained by 10 seconds and is 90% trained by 21 seconds.

xii. Figure 11-b shows that although few new neurons are added after 21 seconds, their centroids are optimized such that the number of cells contributing actively to each control action uses from 1 - 3 near the start to 2 - 4 cells near the 30 - 40 second region. This is the desirable level of neuron cooperation.

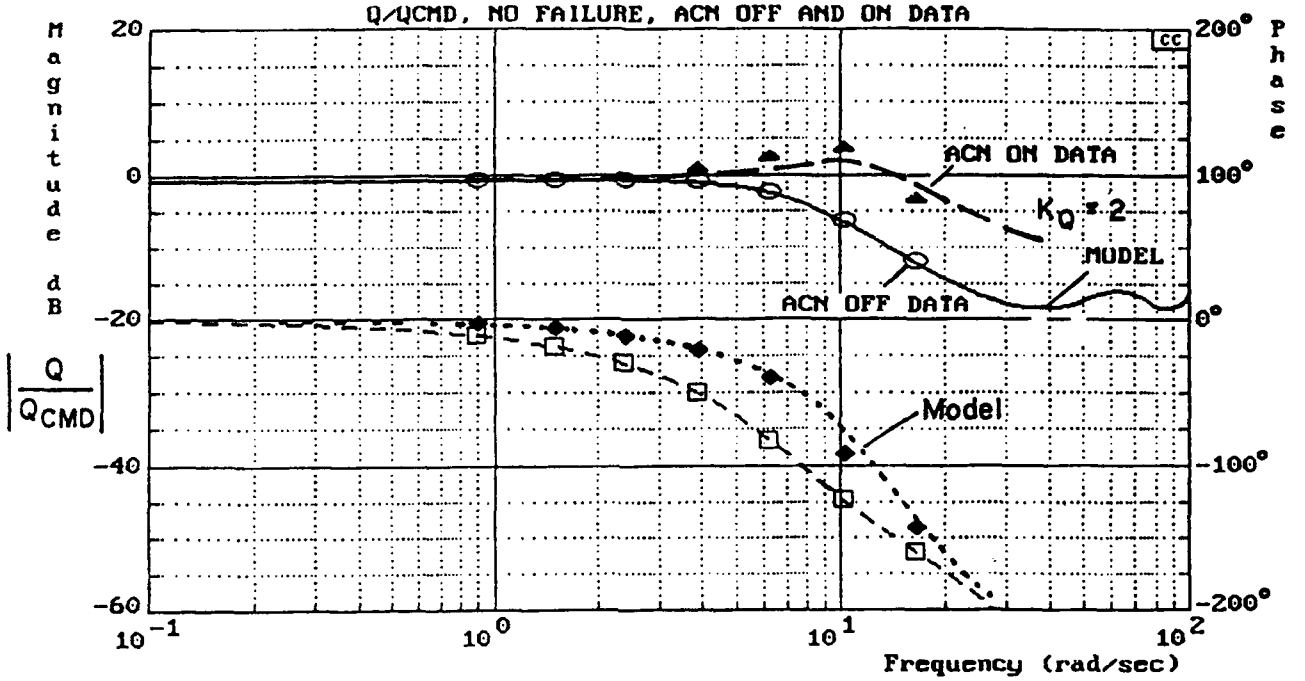
To further validate the dynamic effects of NC Action, describing functions among key signals were computed via STI's FREDA program (see Section III-A). Figure 12 compares the frequency responses for NC = OFF vs ON. While the closed-loop describing function, Q/Q_{CMD} is easy to interpret *per se*, we had also computed the apparent "opened loop" (see Section II-B, Eqn. 7) TF. This interprets NC effects "as if" they were operating on the error alone, and gives insight into how this form of NC is acting. The following conclusions are drawn from Figure 12:

xiii. Because the neurocontroller mimics the feedback-error control signals it "acts as if" it increases the gain of the error loop by a factor of 2, roughly 6 dB. Note that the gain crossover frequency roughly doubles from ω_C = 4.2 to 8.5 rad/sec. This would account for the tighter loop closure and reduced apparent stability margin.

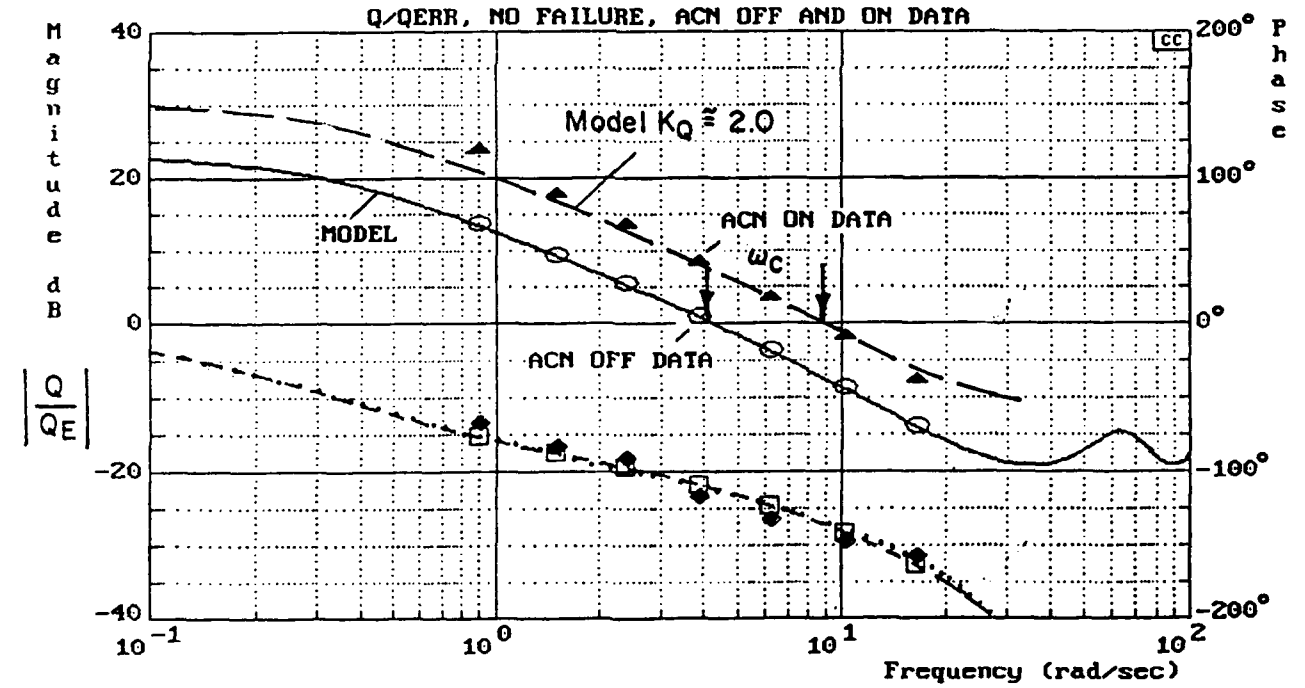
xiv. The closed-loop effective bandwidth (3 dB-down frequency) increases from ω_B = 8 rad/sec to 16 rad/sec with the NC = ON.

xv. There is some increase in closed loop overshoot, suggesting a reduced stability margin.

Because the opened-loop data points with NC = ON appeared as a pure forward loop gain shift of about 6 - 7 dB, the model of Fig. 1 was exercised with K_Q increased from 1.0 to 2.0, the dashed curves of Fig. 12b. The resulting closed-loop "model" is shown as the dashed and dotted lines in Fig. 12-a. The close agreement validates the hypothesis that this NC feedforward loop acts as if it increases the opened loop gain by roughly a factor of 2. Of concern, here, is that the stability



a) Closed-Loop Q/Q_{CMD} Responses, NC OFF and ON



b) Equivalent Opened-Loop Q/Q_{ERR} Responses

Figure 12. Effect of Neurocontroller on Closed- and Opened-Loop Describing Functions

margins, (e.g., gain and phase margins) and closed loop damping ratio are all reduced by this NC external loop, which theoretically should not affect closed loop stability *per se*. Further analysis of this case represents a priority research issue for Phase II.

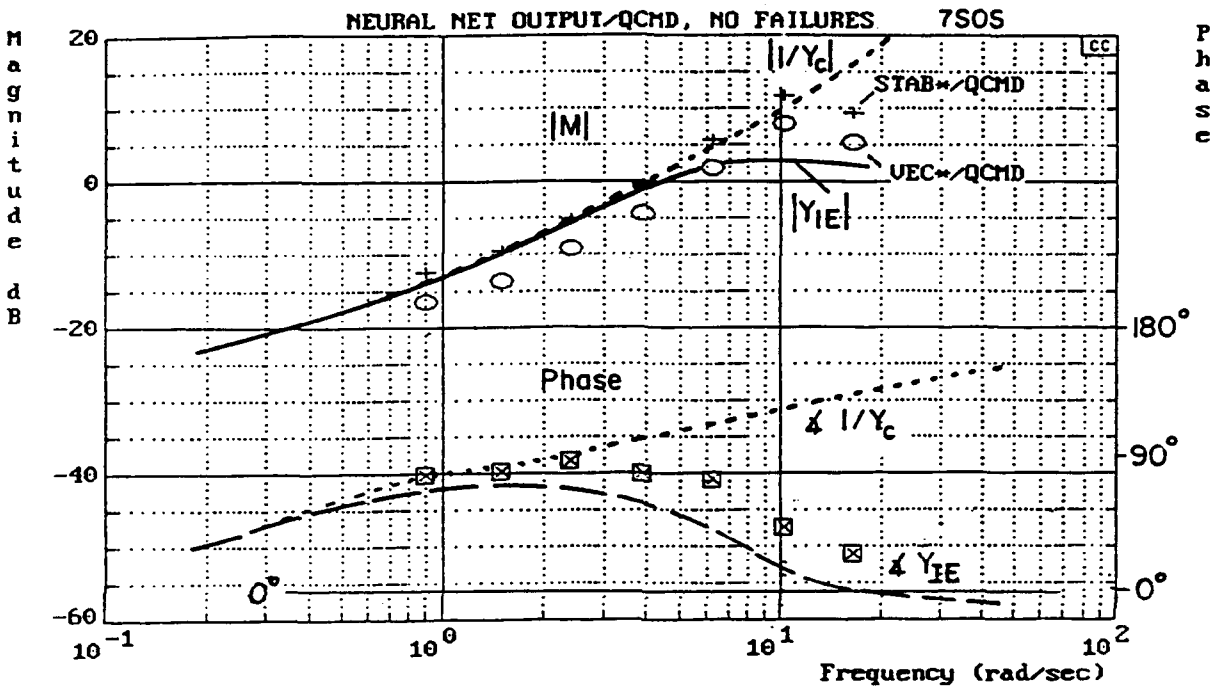


Figure 13. Comparison of Ideal and Measured Feedforward Describing Functions

2. Further Understanding of the Neurocontroller's Feedforward Actions

To further understand the action of the neurocontroller, the apparent command-feedforward describing function of the NC block was computed, i.e., VEC^*/Q_C. Ideally, this should approach the $1/Y_C|_{VEC}$ (s) as shown in Section II-D.

Figure 13 compares the measured data points results with this prediction. It can be seen that the magnitude curve appears close to the desired result at unit frequencies. However, the phase drops off at the higher frequencies providing less lead compensation than required for perfect $1/Y_C$ equalization of Y_C. The reasons were partly predicted in Section II-D, and have the following components:

xvi. The neural-net maps the feedback error's control commands as functions of Q_C, $\dot{Q}_C$, so it can do no better than $Y_E \approx 1/Y_C$. That is, Kawato's Type C scheme can invert the controlled element only as well as Y_{IE} approaches Y_C (i.e., at frequencies well below crossover frequency) per Section II-D. It can be seen that the feedforward phase data points beyond ω_e depart from the $1/Y_C$ curve and approach the Y_{IE} curve.

xvii. There are probably some subtle delay artifacts in mapping the VEC* and STAB* controls vs Q_C and $\dot{Q}_C$, which act to reduce the lead compensation obtainable from serial computations. For example, the $\dot{Q}_C$ state is derived from backdifferencing, so it is about one-half sample delay stale. The ACN mapping and recovery functions may also introduce some delays. Individually such delays are short, but cumulatively they may add up to an appreciable impediment to achieving timely $1/Y_C$ plant inversion.

The overall conclusions of this section are that this neurocontroller/aircraft simulation: works rapidly within an order of magnitude of real-time speeds, improves performance significantly in the no failure case, and can be understood and validated through the combination of the evolved neural-net measures, time histories, and frequency domain describing functions.

B. NEUROCONTROLLER EFFECTS FOLLOWING A THRUST VECTOR FAILURE

To test the ability of the baseline neurocontroller to correct an inside-the-loop control failure, we "failed" the thrust-vector (e.g., as if from a paddle burnout) at 21 seconds into the 42 second run. This is right at the end of the first input epoch, so accurate Fourier analyses of before and after could be obtained. As discussed under Section II-C, this failure does not destabilize the aircraft dynamics but drastically reduces the pitch control effectiveness of the combined STAB and VEC combination. Following failure, the controlled element gain is reduced to about one-third its previous value (1.23 / [1.23 + 2.53] = .33).

Figures 14 and 15 show the time histories of pitch rates and control actions for neurocontrol OFF vs ON, with the VEC failure at 21* seconds, after nearly all unfailed neurons have been clustered by the ACN algorithm. The conclusions drawn from Figs. 14 and 15 are:

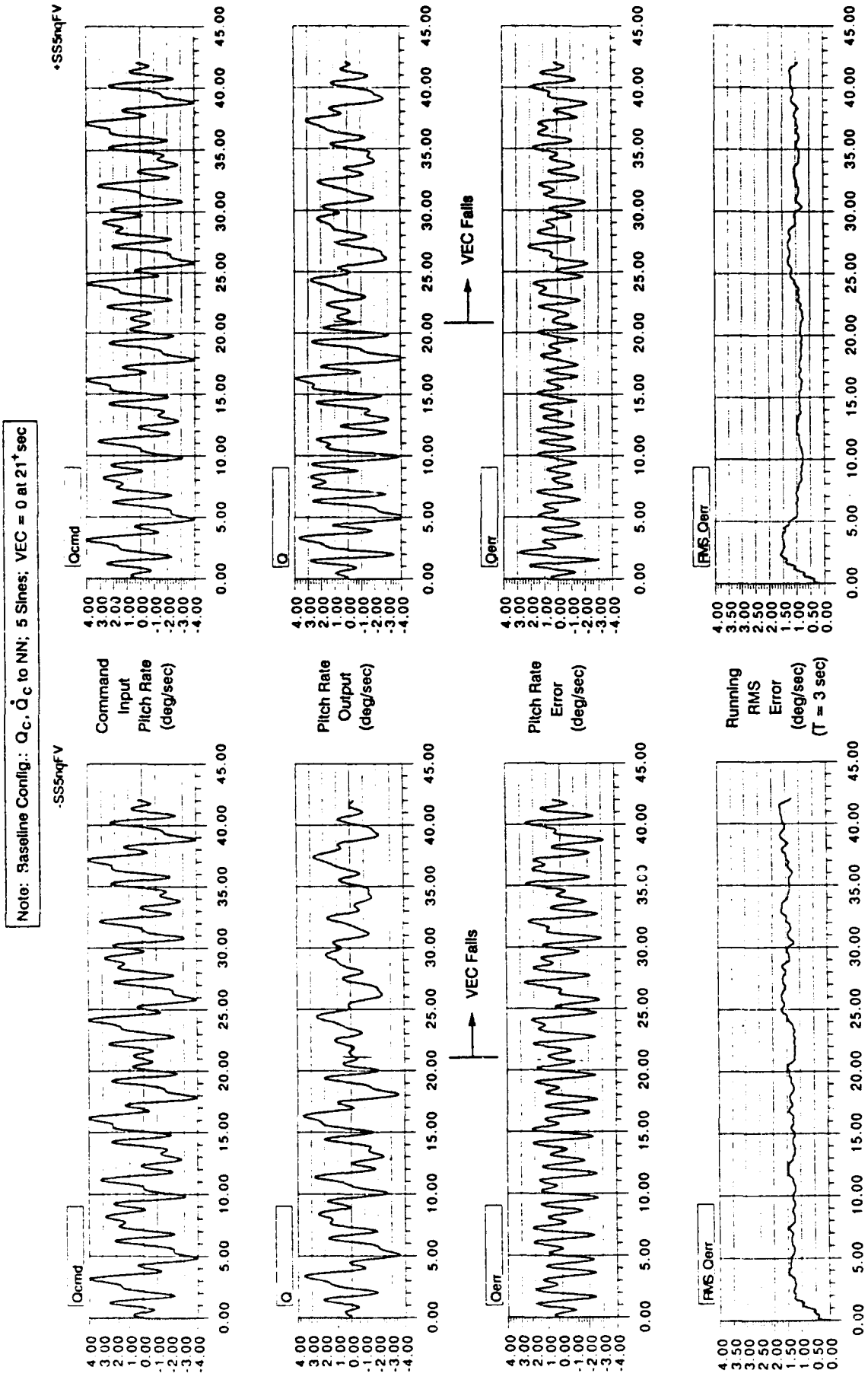
Failure Mode Finding #

i. There is an increase in tracking error (e.g., compare Q and Q_E traces) following a VEC failure. The rms error increases by 15 percent (1.5 / 1.3 deg). With NC = OFF, the output significantly undershoots and lags the command, as expected.

ii. With the NC = ON, there is not much improvement. The failure increases Q_E by 20 percent (1.03/.86). Because the error is larger, the ACN's output is larger; therefore it increases the STAB* commands in proportion to the larger tracking error.

iii. The neurocontroller command to STAB* does not increase by the factor of three needed to restore former closed-loop performance.

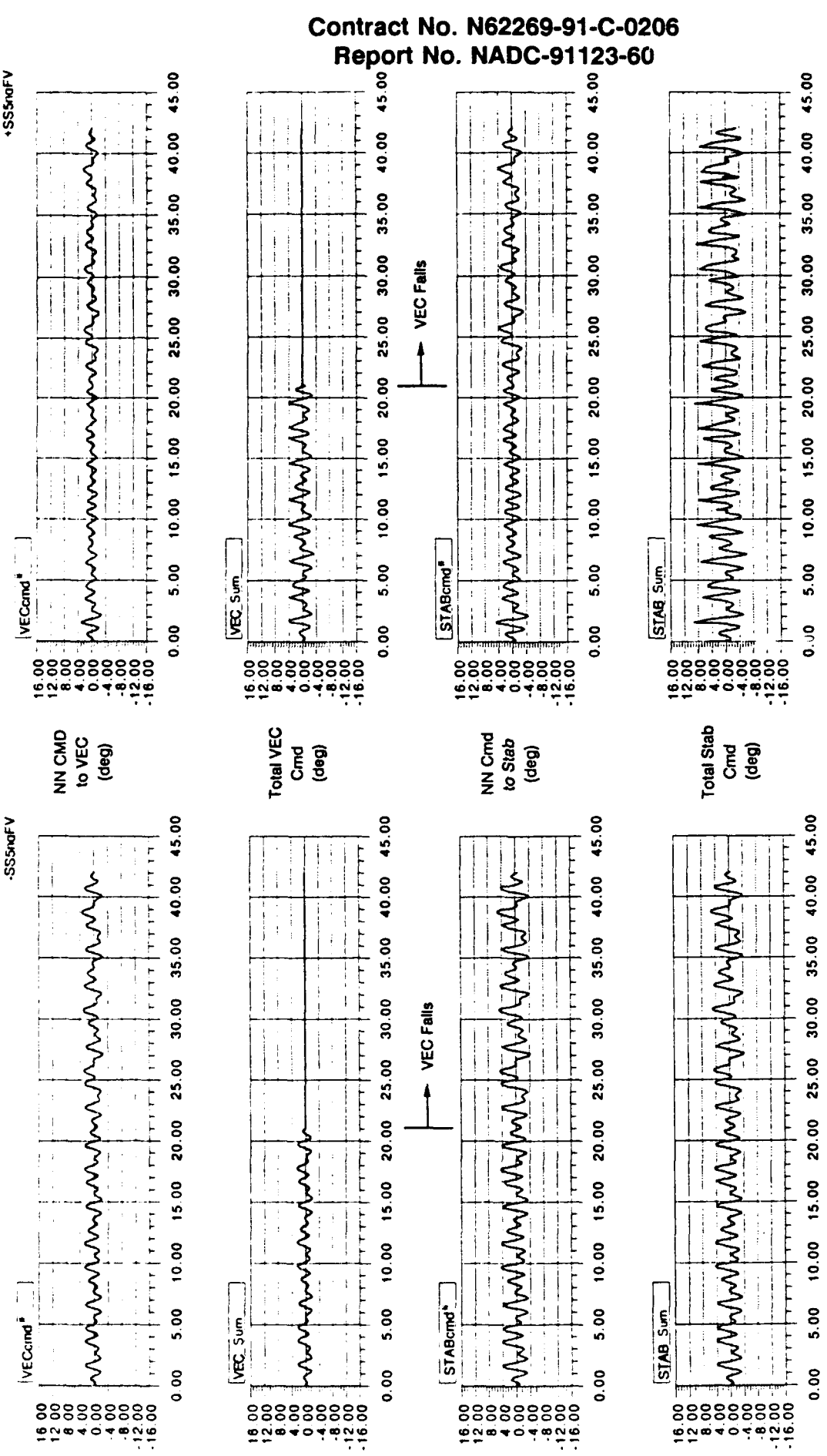
iv. There is no apparent surge of control activity following the failure, as might be expected.



a) NN = "OFF" (feedback error correction only)

b) NN = "ON" (feedback error + NN feedforward)

Figure 14. Effects of Thrust Vector Failure on Pitch Rates (VEC Fails at 21+ Seconds)



b) N/N = "ON" (feedback error + NN feedforward)

a) N/N = "OFF" (feedback error correction only)

Figure 15. Effects of Thrust Vector Failure on Control Signals (VEC Fails @ 21^{+} Seconds)

These findings are consistent with the previous findings that this Kawato Type-C neurocontroller merely maps the unfailed linear controller's control corrections to the command input states. It does not "know" there is a failure, so it just increases the apparent feedback loop's gain, as before.

Figure 16 shows the neural-net activity measures for this failed-VEC case. In accord with the foregoing observations, and even though the ACN learning algorithm is still "ON" after 21* sec, there is no sudden surge of new neuron instantiations or activity to suggest that this baseline ACN is reacting strongly to the failure. Figure 16-c, showing the four new neuron centroids created after the failure as squares, reveals that they are placed near low Q_e and $\dot{Q}_c$ states, where few were placed for the unfailed case (compare with Fig. 11-c). The ACN algorithm maintains a concurrent cell activation at 2 - 4 per correction.

The overall conclusion from this failure experiment is as follows: *because this baseline (Kawato Type C) architecture merely maps the feedback-error-based corrective controls to the neurocontroller transfer functions, it cannot easily learn that one or another control surface has failed, because there are no diagnostic-value states input to the NC.* Thus, it continues to put out corrective control commands to *both* STAB* and VEC* which were learned from the correctly operating feedback system while following the already-mapped command-input states.

This seduced us into adding more states to the neurocontroller input, with little prior analysis of the type in Appendix A.

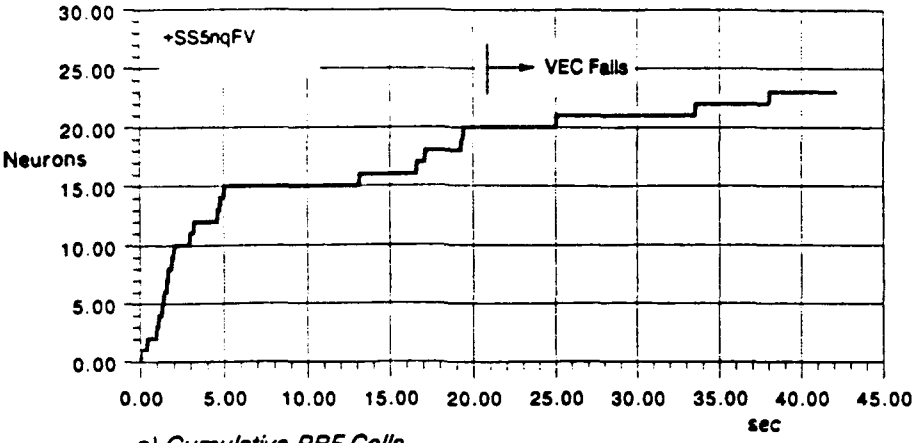
C. EFFECTS OF ADDING OUTPUT STATES TO THE NEUROCONTROLLER INPUTS

It was hypothesized that feeding the desired output states to the NC would help to correct for internal disturbances, including control failures, so a number of runs were made with the output states Q and $\dot{Q}$ added to Q_C and $\dot{Q}_c$ at the neurocontroller input. Somewhat surprisingly, tracking error performance for the unfailed situation was generally not as good as with Q_C and $\dot{Q}_c$ only as the inputs, although there was evidence of better waveform peak matching. The results are shown in Figs. 17 and 18, where the VEC fails at 21* sec, contrasting the baseline and modified cases, but all with NC = ON. The observations from this and similar runs are as follows:

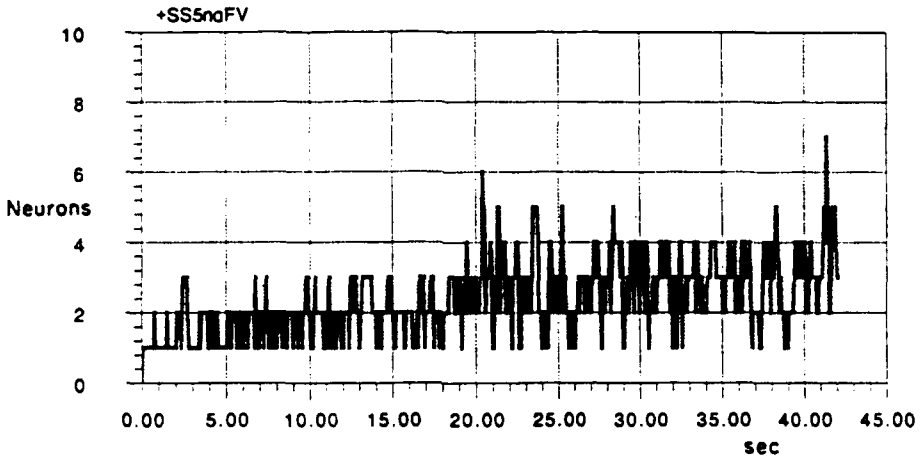
Modified NC Finding #:

i. Considering the *unfailed first 21 sec of* Figure 17 with prior Figure 14, it is apparent that the modified NC input vector gives five percent *worse* rms tracking error (.90 / .86 = 1.05) compared to the baseline case.

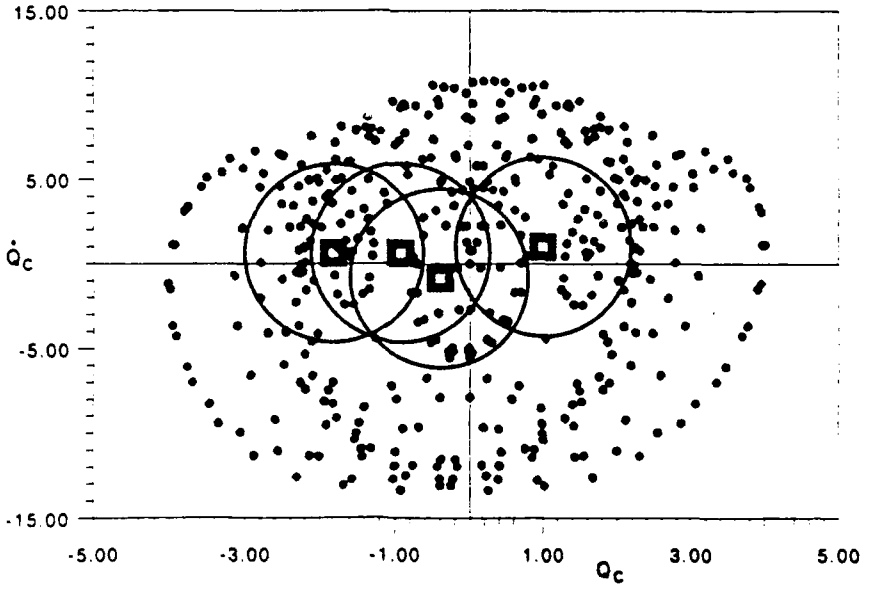
ii. For the post VEC *failure period* (t = 21* to 42 seconds) the modified NC input gives ten percent *worse* rms error (1.1 / 1.0 = 1.10).



a) Cumulative RBF Cells



b) Concurrently Active Cells



c) New Neurons Placed After the VEC Fail at 21+ sec

Figure 16. Neural Control Activity for a VEC Failure at 21+ Seconds

iii. Considering the *control activity* of Fig. 18 for t = 0 - 21 sec, the STAB signal best illustrates that the modified input vector (Fig. 18-b) reduces the extreme peaks (Fig. 18-a), but also introduces some apparent high-frequency "noise."

iv. Following the VEC-fail from t = 21* sec onwards, the modified NC input shows comparable magnitudes to the baseline configuration. The increase of STAB noise is readily apparent in Fig. 18-b near the 35 - 42 second region.

v. The increase in control noise for the modified input vector is probably due to the higher average number of concurrently active cells (e.g., see Fig. 16-c), wherein the changes from, say, 2 to 5 cells active causes jumps in the NN output layer's weighted sum. These appear mostly in the STAB signal, because its actuator has less smoothing effect than the slower VEC servo.

Not too surprisingly, the general conclusion from this and similar runs is that *states which are diagnostic/corrective for inner loop failures must be included in the neurocontroller.* This problem was anticipated and analyzed here, in Appendix - A, and those implications will be further considered in Phase II. The limited Phase I scope prevented exploration of a more complete set of parameters and inputs, but that is easy to do now that the complete simulation has been debugged and runs easily. The complexity of the ACN mapping and the nonlinear dynamics within it implies that a more careful pre-run "systems survey" type of analysis be given to each major NC architectural change. Then, only carefully selected changes should be added gradually, in order to permit understanding at each step. It is all too easy, as we have found out, to mix ever more ingredients into the NC pot, stir wildly, and <u>then</u> try to understand what had been cooked up. A more rational approach will be our discipline in Phase II.

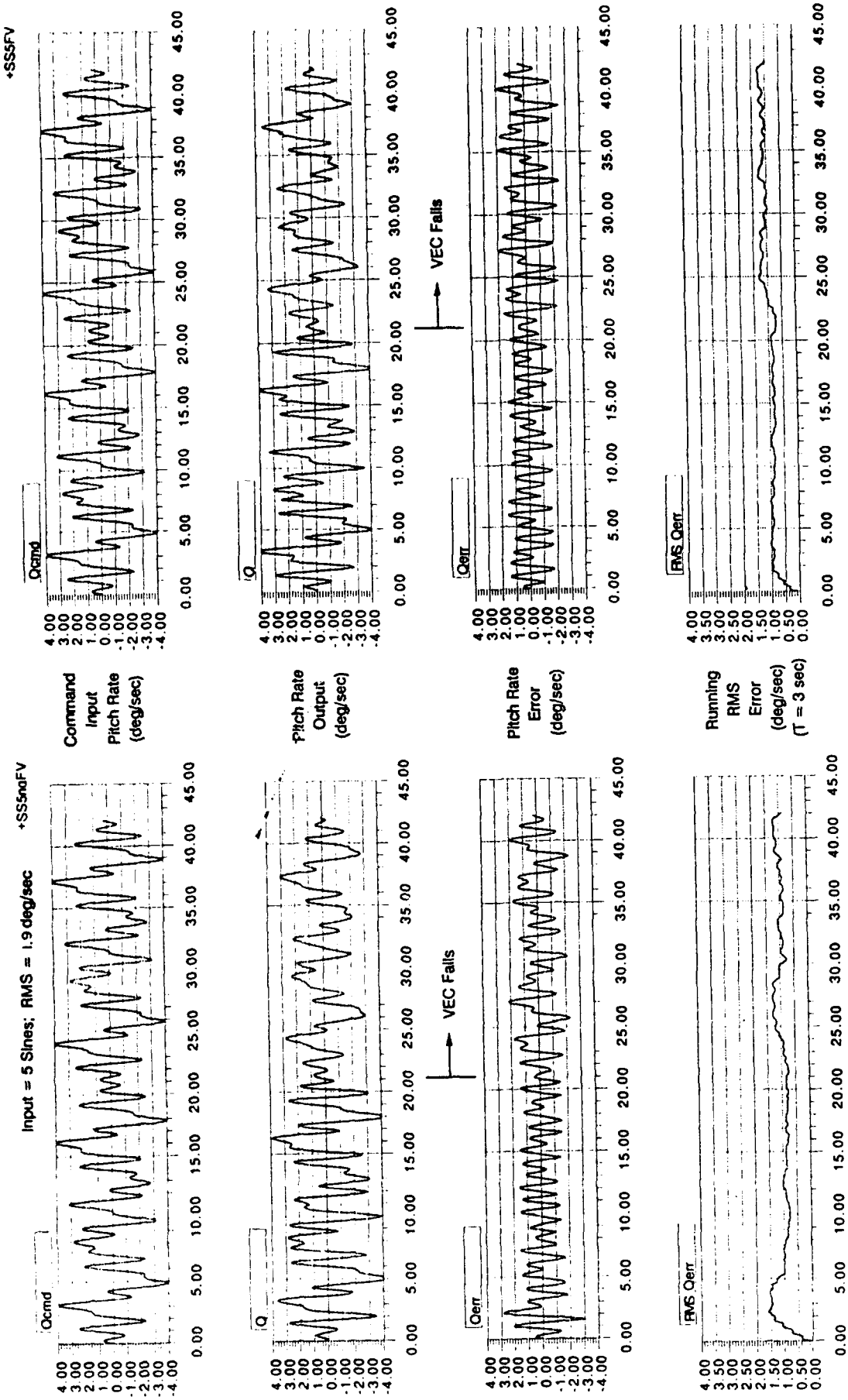
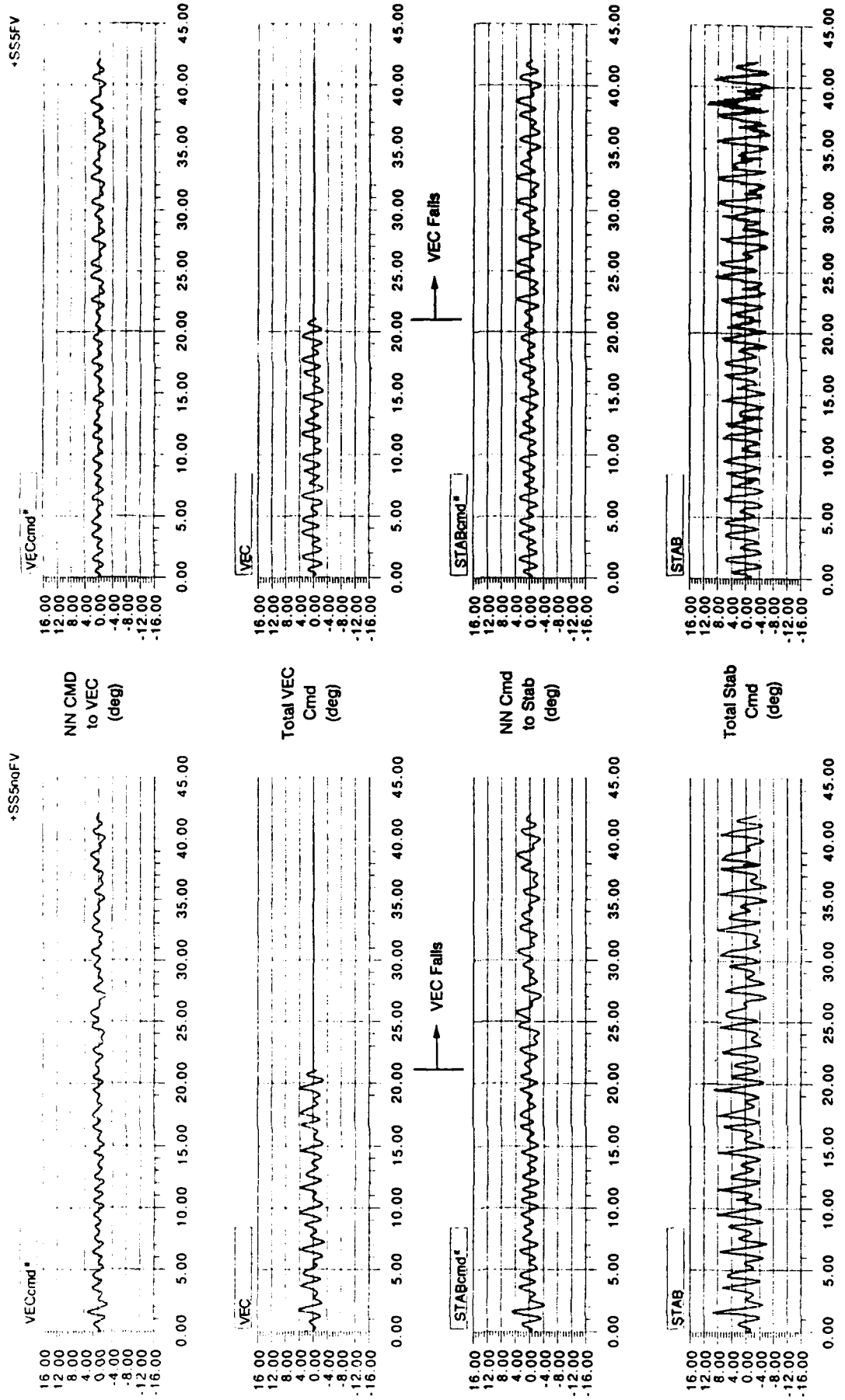


Figure 17. Effects on Pitch Rate of Adding Q and $\dot{Q}$ to Neurocontroller Input



a) Baseline Case: NC = "ON"; Q_C, $\dot{Q}_C$ to NC

b) Modified Case: NC = "ON"; Q_C, $\dot{Q}_C$, Q, $\dot{Q}$ to NC

Figure 18. Effects on Control Signals of Adding Q and $\dot{Q}$ to Neurocontroller Input

VI. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

1. Meeting of Phase I Objectives

During Phase I, most of the revised Objectives stated in Section I were met; specifically:

Conclusion #

i. A feedback learning (Kawato Type C) neural-net controller (NC) was successfully applied to the longitudinal control of an attack fighter via all-movable stabilizer and aft thrust vectoring.

ii. The low-order-equivalent-system (LOES) aircraft system model and all neurocontrol operations were successfully mechanized within STR Corporation's Adaptive Clustering Network (ACN) simulation using the THINK PASCAL 3.0 development environment running on a MacIntosh II-CX computer. Despite the large computing overhead for the development environment, (e.g., debugging routines), the ACN training runs could be made in less than 10 times real-time. The achievement of real-time on-line ACN action now seems feasible on flight control hardware.

iii. In addition to standard time-domain traces of neurocontrol and combined system action, some new measures and methods were demonstrated, such as: use of frequency domain describing functions to unravel the quasi-linear NC actions, neuron centroid overlays on the input state phase plane, and concurrent cell participation in control activity.

iv. The LOES aircraft model evolved in Phase I was actually reduced from more complete 6 DOF and 3 DOF simulations of an actual twin-tailed Navy attack fighter, and is well suited to re-expansion in Phase II. Means for readily validating and exercising the simulation via an efficient quasi-random forcing function were demonstrated.

v. One overall objective of NADC's SBIR solicitation, the detection and reconfiguration of failed aircraft controls was analyzed for the longitudinal case and shown to be a subtle and tough problem for near-co-located controls. Because its solution requires a higher-order aircraft model having additional states and sensors, as well as a multilayered NN, that objective was deferred to Phase II.

vi. A new direction for Phase II emphasis evolved from the various tests and measures for validating, training, and exercising the neurocontrol system. The current lack of standard procedures and measures indicated a clear need for a "Neurocontrol System Evaluation Toolbox" of methods, procedures, forcing functions, analytical strategies, and evaluation criteria.

2. Answering Phase I Questions

Most of the Section I-B specific questions to be answered in Phase I were satisfactorily covered, specifically:

Section I-B Questions

a. The LOES representation of the attack aircraft was shown to be correctly simulated, and the dynamics were representative of a pitch-rate command, attitude-hold SCAS system.

b. The originally proposed neurocontroller architecture (an Intelligent Configuration Management System, ICMS) was examined and deemed too complex for Phase I demonstration. Instead, a type of feed-forward controller, using the feedback error controls to train the ACN, was adopted. This Kawato Type-C scheme mimics a human operator's actions in the so-called "pursuit" level of skill acquisition, and aims to learn the plant inverse, just as a skilled operator does. It promises some novel applications for aircraft control, including learning a rapidly changed or failed control element. In Phase I, we concentrated on the technology for mechanizing, simulating, and measuring such a feedforward/feedback neurocontroller, but did not have time to refine it.

c. Among the diagnostic tests applied productively in Phase I was the frequency-domain describing function of the non-linear, time-varying neural-net input-to-control dynamics. Using this technique we were able to show that the plant inverse could only be crudely, albeit efficiently, approximated by the Kawato Type-C approach. Feeding back the plant outputs to the NC can result in sneak-feedback loops which can destabilize the NC/SCAS/aircraft system. By computing the "effective opened-loop" describing function of the combined feedforward/-feedback system, it was shown that the neurocontroller feedforward roughly doubled the gain of the feedback loop. The closed loop consequences of this effect were reduced stability, poorer damping, and a trade-off of reduced low frequency error for higher frequency overshoot errors.

d. The baseline Kawato Type C system was shown to improve the aircraft's response to controls, resulting in higher apparent pitch-control bandwidth with more overshoot, but reduced rms pitch rate tracking errors (46 percent of the NC = OFF case).

e. The baseline NC does not do a good job of rapidly coping with single-control failures. For example, a likely failure mode — burnoff of a thrust-vector paddle — was not "discovered" for 1 - 2 seconds, and was not restored to more-or-less normal performance until 3 - 5 seconds. This is rapid by some neural-net standards, but not good enough for unstable aircraft undergoing maneuvers. The modifications required for Phase II were clearly revealed in the process of evaluating the Phase I measures and results.

f. Among the "key problems uncovered in Phase I to be solved in Phase II" were the following. (The risk involved is indicated in parentheses):

- Simulation of a much more complete aircraft/SCAS/sensor model within the NC simulation program. (Straightforward).

- Concurrent monitoring for failure and selection of: sensors, signals, criteria for reconfiguration, and on-line adaptive control within the feedforward/feedback scheme adopted here; i.e., multilayer vs. multiple NC's, vs. mode selector, etc. (Challenging).

- Development of a more predictive analytical insight, prior to running the simulations, in order to screen out worthless or unstable concepts. This needs the Neurocontroller Analysis Toolbox. (Needs every multidisciplinary trick in the STI/STR repertoire and some new ones from the NN field; tricky to put in visual and user-friendly form).

- Better criteria for judging the effectiveness, efficiency, and off-design operation of the intrinsically nonlinear neurocontrollers. (Little precedent to go on; straightforward extension of STI Flying Qualities tests and criteria concepts).

3. Some Interesting Findings

Some details on these conclusions should be mentioned, here. In the RESULTS, Section V, there are some 26 specific findings (identified by lower-case Roman numerals) which can be scanned for details but some of the more interesting ones are as follows:

a. Detection of STAB vs VEC Failures (from Appendix A).

The detection and separation of failed elevator (STAB) vs a failed thrust vector (VEC) is a tough problem in longitudinal control, because both are located well aft of the CG and produce roughly comparable ratios of pitching acceleration and normal acceleration at the CG. However, proper placement of the virtual centers of rotation of $\dot{Q}$ and n_z sensors and bandpassing their inputs can probably separate the likely failed surface. This depends on the geometric effective tail arms (ratio of n_z to $\dot{Q}$) being more constant than the actual levels of $\dot{Q}$ and n_z alone, which may vary widely. The wide difference in the response between STAB actuators (high BW) compared with VEC actuators (low BW), allows mid-bandpassed signals to reduce the measurement influence of the former with respect to the latter in a failure detection neural-net. This approach is considered novel and will be a priority objective in Phase II.

b. Inversion of the Controlled Element (From Sections II-D and V-A-1)

The Kawato Type C neurocontroller concept makes use of an ingenious observation to avoid back-propagation-training for inverting the effective controlled element. As shown in Section II-D, if the feedback controller ensemble reduces the closed-loop-error to values very much less than the command, then the control signals generated by the error-feedback loop approximate those required to invert the plant dynamics up to the opened-loop gain crossover frequency, ω_C. These control states are then mapped as a function of the command input and rate, to produce the desired neurocontroller feedforward effect:

$$Y_{NC}(j\omega) = \frac{\text{controls}}{\text{commands}} \approx \frac{1}{Y_C(j\omega)} \qquad \text{for } Y_C \gg 1$$

so that the complete forward loop, Y_{FL} is:

$$Y_{FL} = Y_{NC} \cdot Y_C \approx \left(\frac{1}{Y_C}\right) \cdot Y_C \cong 1.0$$

as desired.

As revealed by our analysis and verified by measurements, the $1/Y_C$ approximation can only be successful for forcing functions bandwidths well below ω_C and for tightly closed but

well-damped feedback loops, a condition that is violated when one or the other control surface fails.

c. Failure Detection and Reconfiguration

The Kawato Type C system cannot produce the correct control actions of the remaining control, nor the shutdown of commands on the failed control that are desired, because no information on control signals within the loop are fed to it. Our attempt to improve this by input of the output states on Q and $\dot{Q}$ to the neurocontroller yielded sneak feedback paths (paralleling but lagging the basic feedback signals) which tightened the loop and destabilized it.

A more sophisticated set of state feedbacks to the neurocontroller is clearly indicated. These must account for not only failures of control surfaces, but also changes in pitch inertia, mass, or dynamic pressure, all of which can be responsible for "confounded" variations in the measured $\dot{Q}$ and n_z signals. We now know what some of the relevant signals are, but need to develop their filtering and pre-NN summing and weighting algorithms.

B. RECOMMENDATIONS FOR PHASE II

The problem area selected in Phase I still is an excellent one for Phase II. The longitudinal control of a tactical fighter using a system with imbedded neurocontrol can demonstrate nearly all of the advantages, problems, and methodology needed for more general cases, while retaining a much simpler model and simulation. Therefore, we should extend the Phase I problem to cover a more realistic aerodynamic model, maneuvers, and failure modes.

Similarly, the Adaptive Clustering Network scheme for the neurocontroller (NC) seems very promising, in that relatively few neurons were required for useful improvements in control. However, adequate and understandable corrections for internal failures and disturbances were marginal with the baseline system which worked off commands and feedback errors only. There were a number of potential NC improvements suggested by the Phase I data, which should be pursued in Phase II.

Finally, in the process of understanding and validating this NC, several new analysis and validation "tools" were shown feasible for analyzing any NC system. Because a comprehensive set of such NC test-and-evaluation tools and procedures is not currently available, we should start the development of such a "Neurocontroller Analysis Toolbox" during Phase II. In almost all of the neurocontrol literature we have read to date, there is little understanding of how to properly *design and validate a complex system*. The steps should include: functional requirements and their implications on design; iterative analysis of different candidates which are not in an optimal-design continuum; multi-factor assessment and evaluation by sometimes non-numerical criteria; and selection of a "best compromise" design. STI has been doing

this for decades in the design of SCAS and AFCS, and we strongly recommend more attention to this art in neurocontroller development.

The specific recommendations are as follows:

1. Complete the Aircraft Model and Kinematics

For Phase I, the complete 3 DOF equations of motion for the tactical fighter were reduced to a 1 DOF set to facilitate simulation. These should be re-expanded to the 3 DOF form, with limits included. Kinematics of the line-of-sight and target error should be added, and sensor locations and dynamics typical of the fighter should be included. The more complex effects of control-effector-hardover or surface-loss failures need to be modeled.

2. Expand the Mission Phase Tasks, Performance Criteria and Implied Forcing Functions

A more comprehensive set of "typical" mission tasks and failure modes needs to be cataloged. A corresponding set of forcing-functions (commands and disturbances) should be derived from the Flying Qualities literature, which are suitable for efficient testing and validation of the neuro-controller/aircraft system. These would include quasi-random inputs (e.g., sum-of-sines-and-cosines) and frequency sweeps (e.g., accelerating chirps), as well as discrete commands and gusts (e.g., steps, "tuned" doublets, and tuned 1-cosine forcing functions). Attention should be given to standard flying qualities test inputs and criteria. Strongly time-varying airspeed maneuvers need to be considered, *such as: weapon release pullups*, Pugachev "Cobra" maneuvers, thrust-onset transients, "scissors," etc. Much of this art has been developed for the U.S. Air Force and Navy by STI, so it is readily available.

3. Review the Current and Potential Modifications to the Kawato Type-C Neurocontroller

We should consider an extension of the current Kawato Type-C neurocontroller (which uses the existing SCAS feedback-error-controls relative to the inputs as an estimate of the plant inverse), which we have come to understand in Phase I. Should NADC desire another NC of comparable complexity to be used, we would consider it early in Phase II.

4. Refine the NC Topology and Processing Element Properties

Some promising additional architectural NC connections and signal shaping ideas were explored late in Phase I. These were intended as refinements of the Phase I effort and should be completed in Phase II. These include:

- Lead-equalization of the derived rates to compensate for computing lags.
- "Barycentric" non-linear distribution of neural parameters (e.g., effective radius), to more closely realize plant inversion, especially at small error conditions. [Coolidge, 1963]

- Parallel NN loops to identify types of failure and redirect (reconfigure) the control commands, (per Appendix A of the Phase I report).

- Different computing rates for the simulations and neurocontroller.

5. Refine the Measurement of Neurocontrol Functioning

One of the gaps in existing NC technology is the lack of well established methods and measures to reveal and validate NC activity and functioning. Some were evolved in Phase I and others were indicated but not tried. These should include: 3D and 4D plots for revealing the ACN neuron distribution and radii in the NN input-parameter space; time-varying NC describing functions in the frequency domain; time variations in the weighting functions; interpretation of the neurocontroller action as effective control loop describing functions; etc. The use of STI's proprietary NIPIP time-domain plant-identification procedure may be of value here, as it has been for measuring human pilots (real adaptive neurocontrollers!) in failure mode control tasks and during the Space Shuttle landing maneuvers. [Hanson and Jewell, 1983]

For use in the NC Toolbox, we should develop a more comprehensive set of standardized and easy-to-simulate forcing functions designed especially to test and evaluate the neurocontroller actions for discrete as well as continuous tasks.

6. Finalize the Aircraft/NC/Measurement Simulation at STR Corp to Include the Foregoing Elements

The final software and hardware platforms for appropriate neurocontroller test cases should be selected. Programming and processor options need to be reviewed. STI and STR Corp should also review the NETSIM environment developed by Draper Labs and used by NADC for concurrent neurocontroller development. Special attention should be given to assessing the suitability of this software for ANN development and as a software platform for simulation experiments involving comparative evaluation of alternative net architectures.

7. Select the Final Neurocontroller Configuration and Tune Its Parameters for Selected Tasks and Failure Modes

The final NC configuration for demonstration of the technology and toolbox application should be chosen, with NADC interaction, and its parameters should be iteratively trained and "tuned," using the NC Toolbox procedures and tests. This NC configuration might not be the optimum with respect to any one (or several) performance criteria, but it would be "typical" and clearly demonstrate the properties of the selected NC and the application of the Neurocontroller Analysis Toolbox.

8. Draft the "Toolbox Architecture"

The tools, measures, and procedures evolved in Tasks 1 and 2 need to be formalized for use in the test and evaluation of a wide variety of neurocontrol systems. We should interact with NADC during Phase II to evolve the overall architecture of a generally usable "Neurocontroller Analysis Toolbox" for testing and analyzing aircraft neurocontrol systems. Adaptation of Toolbox elements to various platforms and portability issues should be reviewed with NADC.

9. Provide a Set of Mission-Phase-Relevant Test Situations and Procedures for the NC Tests

Included should be a matrix of appropriately matched maneuvers, disturbances, failure types, and likely-uncertainty sources for the previously selected mission phases of (say) tactical fighters.

10. Provide the Appropriate Forcing Functions for Each Case in Recommendation 9.

Include: quasi-random and chirp inputs, discrete maneuvers and disturbances, typical failure consequences for a number of tactical situations.

11. Provide the Appropriate Measures and Analysis Procedures for the Above

There is a largely unwritten art in obtaining accurate dynamic and performance measures of such nonlinear systems, much of it pioneered by STI for use in measuring human pilot "neuro-controller" behavior. Guidelines to this art should be systematized and included in Phase II.

12. Provide Appropriate Evaluation Criteria and Current Benchmark Standards

Analyzed tests comprise only half of the evaluation process. Where possible, guidelines to "good" or "adequate" NC performance should be given, or needs for further research clearly indicated. The example case's computation speeds need to be benchmarked for comparison with later improvements. STI experience with Flying Qualities Criteria and Design Guides will be useful, here.

13. Demonstrate the Application of the NC Analysis Toolbox to the Neurocontrol/Aircraft System Developed from Recommendations 1 - 6

At this Phase II stage, the NC Toolbox would not yet be a packaged-deliverable product; that effort properly belongs in Phase III. It would include a rudimentary written Guidelines Manual, with the tests, procedures, and inputs and outputs formally prescribed. Forcing functions and data analysis procedures should be analytically described in pseudocode suitable for coding in most computing platforms or environments. *The Toolbox is not intended to simulate NC systems*, but rather is a set of functions and procedures which, when appropriately connected to an existing simulation, will provide functional metrics for dynamic identification and validation of the system.

We expect that NADC would imbed some of these tests and procedures into their own neural-net control simulation projects and STI/STR would assist in this transfer of technology.

REFERENCES

1. Allen, R.W., and H.R. Jex, "A Simple Fourier Analysis Technique for Measuring Dynamic Response of Manual Control Systems," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-2, No. 5, November 1972.

2. Barto, A., Sutton, and Anderson, "Neuron-Like Adaptive Elements that Can Solve Difficult Learning Control Problems," *IEEE Transactions on Systems, Man, and Cybernetics*, 1983.

3. Bavarian, B., "Introduction to Neural Networks for Automatic Control," *IEEE Control Systems Magazine*, 1988.

4. Coolidge, I.L., *A History of Geometrical Methods*, Dover, New York, 1963.

5. Guez, Allon, Eilbert, and Kam, "Neural Network Architecture for Control, *IEEE Control Systems Magazine*, 1988.

6. Hanson, Gregory D., and Wayne F. Jewell, *Non-Intrusive Parameter Identification Procedure User's Guide*, NASA CR-170398, April 1983.

7. Hecht-Nielson, R., "Counter Propagation Networks," *Proceedings of the International Conference on Neural Networks,* Vol. II, 1987.

8. Hirose, Yoshio, Yamashita, and Hijiya, "Backpropagation Algorithm Which Varies the Number of Hidden Units," *Neural Networks*, Vol. 4, 1991.

9. Holdaway, R., "Enhancing Supervised Learning Algorithms Via Self-Organization," IEEE 2nd International Conference on Neural Networks, 1989.

10. Jex, H.R., "On Improving the Detectability of Aircraft Pitch Control Impairment," WP 1285-1, Systems Technology, Hawthorne, CA, 1991.

11. Jordan, M.I., and Jacobs, "Learning to Control an Unstable System with Forward Modeling," IEEE Conference on Neural Information Processing, Natural and Synthetic, 1989.

12. Kawato, M., Furukawa, K., and Suzuki, R., "A Hierarchical Neural-Network Model for Control and Learning of Voluntary Movement," *Biological Cybernetics*, Vol. 57, 1987, pp. 169 - 185.

13. Kawato, M., "Computational Schemes and Neural Network Models for Formation and Control of Multijoint Arm Trajectory," *Neural Networks for Control*, M.I.T. Press, Cambridge, MA, 1990.

14. Krendel, E.S., and D.T. McRuer, "A Servomechanisms Approach to Skill Development," *Journal of the Franklin Institute*, Vol. 249, pp. 24-42, 1960.

15. McRuer, D.T., I.L. Ashkenas, and D. Graham, *Aircraft Dynamics and Automatic Control*, Princeton University Press, Princeton, NJ, 1973.

16. Musavi, M.T., K.B. Faris, K.H. Chan, and W. Ahmed, "On the Implementation of the RBF Technique in Neural Networks," *JACM*, Vol. 5, 1991, pp. 110 - 115.

17. Narendra, K.S., "Adaptive Control Using Neural Networks," *Neural Networks for Control*, M.I.T. Press, Cambridge, MA, 1990.

Appendix A

Working Paper No. 1285-1

ON IMPROVING THE DETECTABILITY OF AIRCRAFT PITCH CONTROL IMPAIRMENT

June 1991

Henry R. Jex

Contract No. N62269-91-C-0206

A. Introduction

The scenario, here, is a jet-fighter, pulled-up in level flight to about 30 degrees angle of attack (whereby the trim thrust is about one-half of the weight), under an overhead target, with the pilot maneuvering about this trim condition to aim the gunsight. A realistic and challlenging problem case for the application of neural-net detectors and/or controllers in such situations is the need to cope with failure of thrust vector controls during such use. The problem is serious, because from 50 to 100 percent of the aircraft weight may be carried by the vectorable thrust, so an undetected failure (e.g., due to deflector paddle burnaway) could cause loss of the aircraft.

Two failure modes; 1) paddle burnout and 2) stuck or hardover paddles, are the most likely ones, due to paddle operation in a red-hot environment. Both elevator and thrust vectoring will be used synergistically to preserve adequate margins of control effectiveness, and to compensate for the failure or stalling of either one. Furthermore, the center-of-pressure of both elevator and paddle are aft of the center-of-gravity, and this makes difficult the detection of failure of either, as discussed below. This informal paper discusses some strategies for improving the detectability of such subtle failures, so that a neural-net detection system would be easier to train and more positive in operation.

B. Geometry

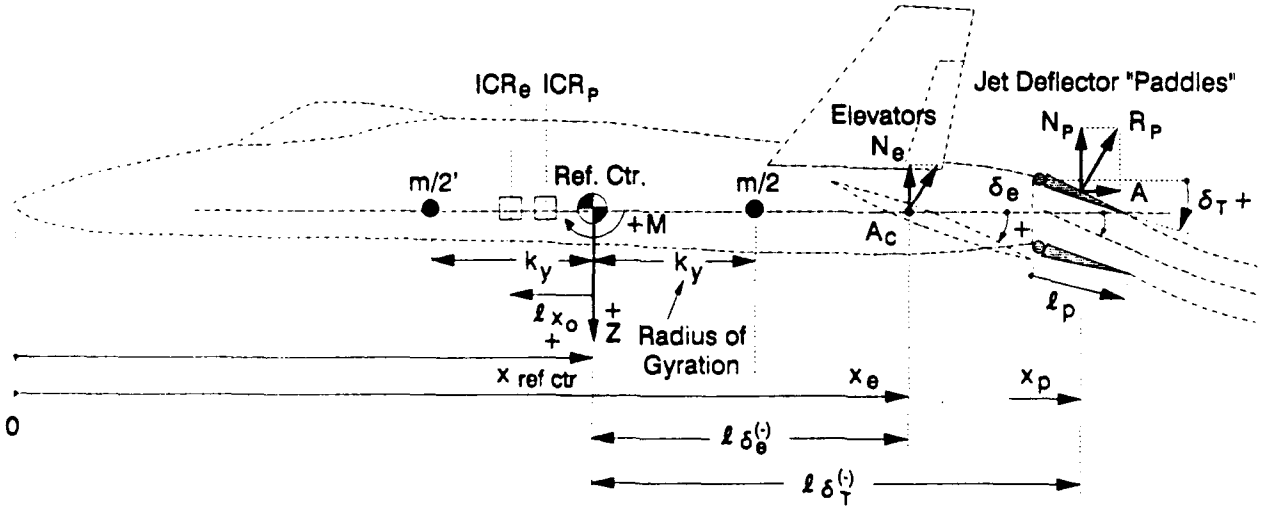


Figure A-1. Definition of Geometric, Inertial, and Aerodynamic Parameters

C. Approach

The basic effect of elevator or thrust deflection paddle deflections is to produce forces acting at x_e and x_p, respectively. The normal components of these forces, N_e and N_p, respectively, produce normal forces and moments at the reference center; for example, the quarter MAC [$\bar{c}/4$] at the Fuselage Reference Line [FRL]. The center of gravity, CG, can move around, so it is common to reference the various aerodynamic forces and moments with respect to this reference CG, then to transfer their moments to the current CG.

When the elevator and thrust paddles have different distances from the reference center, a given normal force by either one will produce a different pitching moment, hence a different angular acceleration about the Reference Center. This, in turn, produces different "instantaneous-centers-of-rotation," ICR_e and ICR_p, respectively.[1] Two normal accelerometers in the vicinity of these ICRs (say, one at the Ref. Ctr. and one at some distance $l_x \sim k_x$ ahead of it) can detect the different effects of these different ICRs and can permit identification of changes in either control's integrity. Alternately, one normal accelerometer can provide the same information, provided that the body bending between the sensors and ICRs is minimal in either case.

An artificial neural net, ANN, working on the outputs of such properly located sensors should be able to detect and measure quickly the impairment to either control device and to start corrective reconfiguration of the controller.

If the elevator action and thrust-paddle action both take place at the same distance from the Ref. Ctr., it is theoretically impossible to separate out which is impaired, if both move in synchronism. If one (say the paddle) lags the other, then it may still be possible to separate their effects. If the two surfaces are separated, but moved synchronously, it is more difficult to separate impairments, but still possible. If both surfaces are impaired concurrently and proportionally, it is very difficult to detect which is at fault, because the absolute angular acceleration is, at any given moment, a non-robust parameter due to uncertainties in CG, dynamic pressure, and aerodynamic effectiveness. On the other hand, the relative ICR locations are dependent on locations of the effective control forces, and these locations are less uncertain than the components of the forces themselves. Thus, the ANN sensor complex and processing should relate to ICR locations. This note will cover the theoretical principles and considerations.

D. Theory

Refer to Fig. 19, using conventional aircraft body-centerline axes for simplicity: Let η_i be the fraction of original elevator or paddle area (S_{io}) remaining ($0 \leq \eta_i \leq 1.0$).

[1] The instantaneous center of rotation is readily experienced while batting a baseball. If the ball impacts the bat at its "sweet spot" (label), the ICR is at the handle, so no impact is felt. Other impact points are strongly reflected in the perceived impact at the handle.

Pitch moment of inertia: $I_{yy} = m\, k_y^2$ (slug - ft^2)

Z Force from + elevator: (Z is + dn; C_N is + up)[2]

$$Z(\delta_E) = -\eta_E S_{E_o} q_E C_{N\delta_e} \delta_E \quad (1)$$

Moment from + elevator: (M is + nose up)

$$M(\delta_E) = Z(\delta_E)\, l_{\delta_e} \quad (2)$$

$$= -\eta_E l_{\delta_e} S_{E_o} q_E C_{N\delta_e} \delta_E \quad (3)$$

Recall that the conventional "dimensional derivatives" are defined as the linear or angular accelerations from control deflection:

Linear Acceleration:

$$Z_{\delta_e} \cdot \delta_e = \frac{Z(\delta_e)}{m} = \frac{-\eta_e S_{e_o} q_e C_{N_{\delta_e}} \delta_e}{m} \quad (4)$$

Rotational Acceleration:

$$M_{\delta_e} \cdot \delta_e = \frac{M(\delta_e)}{I_y} = \frac{-\eta_e l_{\delta_e} S_{e_o} q_e C_{N_{\delta_e}} \delta_e}{m k_y^2} \quad (5)$$

Recall that, at the Instantaneous Center of Rotation (ICR, at l_{x_o} ahead of the CG), the net normal acceleration from translation and rotation due to a step δ_E is zero.[3]

$$\ddot{Z}_0(\delta_e) = 0 = Z_{\delta_e} - l_{x_o} M_{\delta_e} \delta_e$$

so:

$$l_{x_{\delta_e}} = \frac{+Z_{\delta_e}}{M\delta_e} \frac{\delta_e}{\delta_e} = \frac{+Z_{\delta_e}}{M_{\delta_e}} \quad (6)$$

[Note: because Z_{δ_e} is (-), M_{δ_e} is (-), then l_{x_o} is (+), ahead of CG)

Further, expanding out the components of each term:

$$\left[l_{x_{\delta_e}} = \frac{(-\eta_e S_{0e} q_e C_{N_{\delta_e}} \delta_e)/m}{(-\eta_e S_{0e} q_e C_{N_{\delta_e}} \delta_e)/m k_y^2} = \frac{k_y^2}{l_{\delta_e}} \right] \quad (7)$$

or,

$$l_{x_o} \cdot l_{\delta_e} = k_y^2, \quad (8)$$

[2] $+\delta_e$ is a control rotation (vector) parallel to the +Y axis (trailing-edge down)

[3] McRuer, Ashkenas and Graham, *Aircraft Dynamics and Automatic Control*, Princeton University Press, 1973, (pp. 452 and 486).

the well-known relationship between the percussion point, instantaneous-center-of-rotation, and radius-of-gyration, k_y,[3]

$$(k_y^2 = I_y/m)^{[3]} \tag{9}$$

Similarly, the equations of motion for the thrust paddle are given by the forces due to thrust deflection, δ_T; where the exit thrust is T:

$$\ddot{Z}(\delta_T) = T\sin\delta_T \doteq T\delta_T^r \qquad (\delta_T \text{ in rad}) \tag{10}$$

$$M(\delta_T) = l_{\delta_T} \cdot T \cdot \delta_T \tag{11}$$

And, by analogy with the above relationships:

Normal Acceleration:

$$Z_{\delta_T} \cdot \delta_T = \left(\frac{T}{M}\right) \delta_T \tag{12}$$

Angular Acceleration:

$$M_{\delta_T} \cdot \delta_T = \left(\frac{l_{\delta_T} T}{m k_y^2}\right) \delta_T \tag{13}$$

ICR loci:

$$\left[l_{x_{cr}} = \frac{Z_{\delta_T}}{M_{\delta_T}} = \frac{k_y^2}{l_{\delta_T}} \right] \tag{14}$$

Because, (for these fighters), l_{δ_T} is larger than l_{δ_E}, the thrust ICR, $l_{x_{cr}}$, will be closer to the CG than $l_{x_{cE}}$.

Numerical Values for the Fighter Example

We will use values for a typical twin-tail fighter case in [4]; at a high trim angle of attack, (α = 30°) and low speed ($\overline{V}$ = 220 ft/s). For this case thrust vectoring is important, and the failure mode of thrust deflector burnout is critical. Numerical parameters are given in Table 1.

Then, using Eqn. 14 to find $l_{x_{cr}}$ and Eqn. 8 to find l_{δ_E}, and comparing the elevator and thrust deflection parameters, the parameters of Table 2 result.

[4]Myers, T.T., B.L. Aponso, and Z. Parseghian, *Task Tailored Controls, Volume II: Enhanced Fighter Maneuverability Technical Report* NASA/Langley, NAS 1-18661-2, September 1987.

TABLE 1 Fighter Example Pitch Axis Parameters at $\hat{\alpha}_{trim}$ = 30°

Item		English Units		SI Units
Wing Area	S =	400* ft^2	=	37.17 m^2
Wing Chord	c =	11.52* ft	=	3.51 m
Weight	W =	32,366* lb	=	143,964 newtons
Mass	m =	1006.1 slug	=	14,683 kg
Pitch Inertia	I_y =	123936* sl ft^2	=	168,032 kg m^2
Radius of Gyration, pitch	k_y =	$\left[\frac{123936}{1006.1}\right]^{(1/2)}$ = 11.10ft	=	3.38 m
Z_{acc} due to elevator	Z_{δ_e} =	-13.1 ft/sec^2/rad	=	-3.99 m/sec^2/rad
Z_{acc} due to thrust deflection	Z_{δ_T} =	-16.1 ft/sec^2/rad	=	-4.91 m/sec^2/rad
Pitch Accel. due to elevator	M_{δ_e} =	-1.23 rad/sec^2/rad	=	-1.23 rad/sec^2/rad
Pitch Accel. due to thrust deflection	M_{δ_T} =	-2.53 rad/sec^2/rad	=	-2.53 rad/sec^2/rad

[* = given, rest are derived]

TABLE 2 ICR's and Effective Control Arms

	Elevator Deflection	Thrust Deflection
Inst. Ctr. Rt'n.: $l_{x_\delta} = \frac{Z_\delta}{M_\delta}$ (+ ahead of CG)	$l_{x_{\delta_e}} = \frac{-1.31}{-1.23} = +10.65$	$l_{x_{\delta_T}} = \frac{-16.1}{-2.53} = +6.36$ ft
Effective Control		
Moment Area: $l_\delta = \frac{k_y^2}{l_x}$ (+ aft of CG)	$l_{\delta_e} = \frac{11.1^2}{10.65} = 11.6$ ft	$l_{\delta_T} = \frac{11.1^2}{6.36} = 19.4$ ft

On a sketch of the fighter, the effective centers are roughly as noted below:

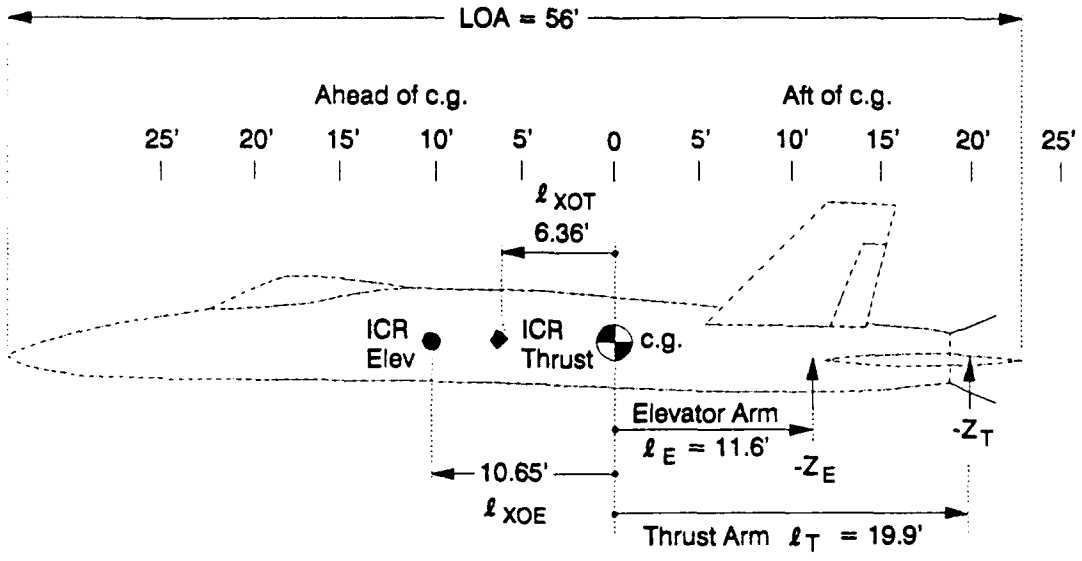


Figure A-2. Locations of Effective Control Arms and Instantaneous Centers of Rotation

Note that the effective moment arms are not precisely at the geometric surface centers due to aerodynamic interactions with the fuselage, etc. Nevertheless, it is found from experience that these geometric arms or locations are more stable values compared to the wide ranges in their component terms (e.g., q) over the flight regime, so their estimate by suitably placed sensors is the robust way to detect failures.

Implications for Detection of Elevator vs Thrust Deflector Failures

1. The ratio of CG accelerations due to elevator/thrust deflections is (from Table 1) = -13.1/-16.1 = .81. This .19 percent difference is not much, compared with the uncertainties present at any flight condition. This means that a CG mounted accelerometer would not be very useful alone in separating the differences, but is essential in detecting ICR or l_δ variations, with the help of angular accelerations.

2. The ratio of pitching angular accelerations for elevator/thrust deflections is (from Table 1) -1.23/-2.53 = .49, a roughly 1:2 ratio or a 51 percent difference. Therefore, angular acceleration per control deflection is a more sensitive discriminant of thrust deflection vs elevator deflection, but this is still not robust across flight conditions.

3. From Table 2, the ICR for elevator is nearly 11 ft vs 6 ft for the thrust deflection, and this will be similar over a wide range of flight conditions. The ICR is measured by the ratio of linear to rotary acceleration (Eqn. 6) and is roughly independent of flight conditions

or control deflections. So, the ratio of CG acceleration to rotary acceleration (derivative of pitch rate) at frequencies where control surface effects are dominant, is an excellent parameter to use for detection of failure. Bending mode effects must be filtered out or compensated for.

$$I_{z_{\alpha}}(j\omega) = \frac{\ddot{Z}_{CG}}{\ddot{\theta}_{CG}}(\delta_i) = \frac{\ddot{Z}_{CG}}{\dot{q}_{CG}}(\delta_i)$$

The frequency range where this is valid lies just beyond the short-period mode, where the M_{δ_e} or M_{δ_T} term dominates the acceleration and angular response (rigid-body, high-frequency asymptote).

4. There is another factor to be considered: the difference in the servo bandwidth for the elevator motions vs thrust-deflector motions. For the test case Ref. [3] gives:

$$1/T_e = 1/.032\text{sec} = 31.25 \text{ r/s}$$

$$1/T_{E_T} = 1/.12\text{sec} = 8.33 \text{ r/s}$$

This is a 4:1 range, and if both surfaces were commanded to oscillate at about 2.5 Hz (16 r/s) the elevator would respond nearly at the command amplitude, while the thrust deflector would only oscillate at about one-quarter of the command. This implies that band pass filtration of the accelerometers would be a fruitful way to further discriminate between elevator and deflector effects. The bandpass would, ideally, be in the region above the aircraft closed loop short period (here, in the order of .8 rad/sec) and the lowest servo bandwidth (about 8 rad/sec), i.e., in the range of about 4-8 rad/sec, or about 1 Hz.

Conclusions

The best signals to feed to the neural-net ensemble, in order to most clearly detect failures in elevator vs thrust deflection are:

1. CG acceleration, in the passband near 1 Hz ± .5 Hz
2. pitch acceleration, in the passband near 1 Hz ± .5 Hz.

Pitch rate is not as good a signal as pitch acceleration for the reasons analyzed in the text. A neural net might learn (configure itself) to give a virtual lead-lag on pitch rate, thereby yielding a pseudo-acceleration signal. However, it should be quicker to train with angular acceleration than rate. In the actual case, the actual bending mode effects on the sensors must also be considered.

The neural net should be trained to recognize the relative (ratio) level between these (and any other consistent but subtle clues) to identify the failure of one or the other surface.

Attn: Dr. K. Narendra

Dept. of Elec. & Comp. Engr.....1
University of New Hampshire
Durham, New Hampshire 03824
Attn: Dr. T. Miller

Dept. of Elec. Engr.....1
University of Southern California
Los Angeles, CA 90089-0272
Attn: Dr. B. Kosko

Dept. of Comp. & Information Sciences.....1
University of Massachusetts
Amherst, MA 01003
Attn: Dr. A. Barto

Dept. of Brain & Cognitive Sciences.....1
Massachusetts Institute of Technology
Cambridge, MA 02139
Attn: Dr. M. Jordan

Dept. of Elec. & Comp. Engr.....1
Drexel University
Philadelphia, PA 19104
Attn: Dr. A. Guez

NAVAIRDEVCEN.....55
Attn: Library 8131 (5)
Attn: Mr. M. Steinberg 6012 (50)

Chatsworth, CA 91311
Attn: Dr. C. Lin

Accurate Automation.....1
1548 Riverside Dr.
P.O. Box 11295, Suite B
Chattanooga, TN 37406
Attn: Mr. R. Pap

Robicon Systems.....1
301 N. Harrison St.
Suite 242
Princeton, NJ 08540
Attn: Dr. S. Lane

Applied Physics Lab.....1
Johns Hopkins University
Johns Hopkins Road
Laurel, MD 20707
Attn: Mr. David Yost

Dept. of Mechanical & Aero Engineering.....1
Arizona State University
Tempe, AZ 85287-6106
Attn: Dr. Dave Schmidt

School of Aeronautics and Astronautics.....1
Purdue University
West Lafayette, IN 47907
Attn. Dr. D. Andrisani

School of Aerospace Engineering.....1
Georgia Institute of Technology
Atlanta, GA 30332
Attn: Dr. A. Calise

Dept. of Aeronautics and Astronautics.....1
Stanford University
Stanford, CA 94305
Attn: Dr. Robert Cannon

The University of Kansas.....1
2004 Learned Hall
Lawrence, KS 66045
Attn: Dr. J. Roskam

Dept. of Mech. & Aerospace Engr.....1
Princeton University
Princeton, NJ 08544
Attn: Dr. Robert Stengel

Center for Systems Science.....1
Yale University
New Haven, CT 06520

Schoephoester Road
Winsor Locks, CT 06096

General Electric.....1
Aerospace Control Systems
P.O. Box 5000
Binghampton, NY 13902
Attn: Mr. R. Quinlivan

General Electric.....1
P.O. Box 8
Schenectedy, NY 12301
Attn: Mr. T. Brownell(Bldg. KWD, rm. 206)

Smiths Industries.....1
Electronic Systems & Software Development
4141 Eastern Ave, S.E.
Grand Rapids, MI 49518-8727
Attn: Mr. M. Bird

Coleman Research Corp.....1
9302 Lee Highway
Fairfax, VA 22031
Attn: Mr. Q. Lam

Harris Corp.....1
Government Aerospace Systems Division
P.O. Box 9400
Melbourne, FL 32902

Charles Stark Draper Laboratories.....2
555 Technology Square
Cambridge, MA 02139
Attn: Mr. W. Baker
Attn: Mr. J. Farrell

Guided Systems Technology.....1
430 10th St, NW, Suite N 107
Atlanta, GA 30318-5769
Attn: Dr. E. Corban

Scientific Systems.....1
500 W. Cummings Pk.
Suite 3950
Woburn, MA 01801
Attn: Mr. R. Mehra

STR Corporation.....1
10700 Parkridge Boulevard
Reston, VA 22091-4356
Attn: Mr. R. Walters

American GNC Corporation.....1
9131 Mason Ave.

Bell Helicopter, Textron.....1
P.O. Box 482
Ft. Worth, TX 76101

Honeywell Systems & Research Center 1
3660 Technology Drive
Minneapolis, MN 55418
Attn: Mr. Thomas Cunnungham

Honeywell, Inc.....1
Defense Avionics System Division
9201 San Mateo, NE
Albuquerque, NM 87113-2227
Attn: Mr. Peter Briggs

Collins Rockwell International.....1
400 Collins Road, N.E.
Cedar Rapids, IA 52406

GEC Avionics.....1
1375 Kettering Tower
Dayton, OH 45423

Lear Siegler, Inc.....1
Astronics Division
3400 Airport Ave.
P.O. Box 442
Santa Monica, CA 90406
Attn: Mr. Dale Uyeda

Allied Signal 1
Bendix Flight Systems
43 Williams Ave.
Teterboro, NJ 07608

Systems Control Technology, Inc.....1
2300 Geng Rd
P.O. Box 10180
Palo Alto, CA 94303-0888
Attn: Mr. J. Vincent

Barron Associates, Inc.....1
Route 1
Box 159
Stanardsville, VA 22973-9511
Attn: R. Barron

Calspan Corporation.....1
Flight Research Department
Box 400
Buffalo, NY 14225

Hamilton Standard.....1

14543 SE 51st Street
Seattle, WA 98006

Boeing Helicopters.....1
P.O. Box 16858
Philadelphia, PA 19142
Attn: Mr. B. McManus

Boeing Helicopter Computing.....1
P.O. Box 33126
Philadelphia, PA 19142
Attn: Mr. D. Rock(M/S P30-58)

Boeing Military Airplane Company.....1
P.O. Box 7730, M/S:K75-65
Wichita, KS 67277-7730

Boeing Defense & Space Group.....1
P.O. Box 3707
Seattle, WA 98124
Attn: Mr. L. Dailey(M/S 4C-70)

Rockwell International.....2
North American Aircraft Operations
P.O. Box 90298
Los Angeles, CA 90009
Attn: Mr. R. Schwanz (MS GB13)
Attn: Mr. D. Moore(MS 011-GB24)

Rockwell International Science Center.....1
1049 Camino Dos Rios
P.O. Box 1085
Thousand Oaks, CA 91358
Attn: Dr. S. Chand

General Dynamics Corporation.....1
Ft. Worth Division
P.O. Box 748
Ft. Worth, TX 76108
Attn: Mr. C. Droste

General Dynamics Pomona Division.....1
P.O. Box 2507
Pomona, CA 91769
Attn: Mr. Walter Waymeyer

Sikorsky Aircraft.....1
North Main Street
Stratford, CT 06602

Kaman Aerospace Corp.....1
P.O. Box 2
Bloomfield, CT 06002

Attn: Mr. G. Vetch
Attn: Mr. H. Harshburger
Attn: Mr. J. Urnes
Attn: Mr. S. Aylward
Attn: Mr. P. Cheng

McDonnell Douglas Helicopters.....1
5000 E. McDowell Road
Mesa, AZ 85205-9797
Attn: Mr. S. Osder

Douglas Aircraft Company.....1
3855 Lakewood Blvd.
Long Beach, CA 90846
Attn: Mr. John D. McDonnell

Systems Technology, Inc 1
13766 S. Hawthorne Blvd.
Hawthorne, CA 90250

Northrop Corp.....1
Aircraft Group
One Northrop Ave.
Hawthorne, CA 90250-3277
Attn: Mr. P. Shaw

Lockheed Georgia Co.....1
Marietta, GA 30063
Attn: Mr. W. Hargrove

Lockheed Research Laboratory.....1
3251 Hanover St.
B284-09110
Palo Alto, Ca 94304
Attn: Mr. T. Washburne

Grumman Aerospace Corp.....1
Bethpage, NY 11714
Attn: Mr. R. Gran (M/S A08-35)

Grumman Corporate Research Center.....1
Bethpage, NY 11714-3580
Attn: Dr. C. Huang

Boeing Aerospace & Electronics.....1
P.O. Box 3999
Seattle, WA 98124-2499
Attn: Q. Mendoza (MS 9Y-18)

Boeing Aircraft Co.....1
P.O. Box 3707
Seattle, WA 98124

Boeing Military Airplane Co.....1

Director, NASA Lewis Research Center.....1
21000 Brookpark Road
Cleveland, OH 44135
Attn: Dr. S. Garg(MS 77-1)

Director, Johnson Space Center.....2
Houston, TX 77058
Attn: Mr. K. Cox
Attn: Dr. R. Lea(PT 4)

Director,.....1
NASA Ames-Dryden Flight Research Facility
P.O. Box 273
Edwards, CA 93523-5000
Attn: Mr. D.A. Deets

Director, NASA.....2
Langley Research Center
Hampton, VA 23365
Attn: Mr. J. Creedon
Attn: Mr. Greg Walker(M/S 286)

Director, Applied Technology Laboratory.....1
U.S. Army RTL (AVCOM)
Fort Eustis, VA 23604-5577
Attn: Mr. J. Stephens (SAVRT-TY-ASI)

Commander, Aviation Applied Technology Directorate.....1
U.S. Army Aviation Systems Command
Fort Eustis, VA 23604-5577
Attn: Mr. J. MacDonald(SAVRT-TY-ASI)

U.S. Army Missile Command.....1
Redstone Arsenal, AL 35898
Attn: Mr. K. Grider(AMSMI-RD-SS, Bldg. 5400)

Director, AFWAL, WPAFB.....5
Dayton, OH 45433-6553
Attn: Mr. D. Rubertus (FIGL)
Attn: Mr. M. Mears (FIGL)
Attn: Mr. C. Dyer (FIGX)
Attn: Mr. P. Chandler(FIGX)
Attn: Dr. M. Pachter(AFIT/ENG)

National Science Foundation.....1
Division of Emerging Engineering Technologies
1800 G. St. ,N.W.
Washington, D.C., 20550
Attn: Dr. P. Werbos

McDonnell Aircraft Co.....5
P.O. Box 516
St. Louis, MO 63166

DISTRIBUTION LIST

No. of Copies

Administrator, Defense Technical Info. Center.....2
Bldg. No. 5, Cameron Station
Alexandria, VA 22314

Director, Office of Naval Technology.....1
800 N. Quincy St.
Arlington, VA 22217-5000
Attn: Mr. W. King

Commander, Naval Air System Command.....4
Department of the Navy
Washington, DC 20361
Attn: Library (5004) - (2)
Attn: Mr. J. Rebel (AIR-5301)
Attn: Mr. H. Agnew (AIR-53014)

Chief, Office of Naval Research.....1
800 N. Quincy St.
Arlington, VA 22217-5000

Superintendent, Naval Postgraduate School.....1
Monterey, CA 93940
Attn: Dr. D. Collins

Naval Avionics Center.....1
Indianapolis, IN 46218

Naval Air Test Center.....2
Patuxent River, MD 20670
Attn: Mr. J. Darling (SA-40)
Attn: Mr. R. Burton (SA-100)

Naval Weapons Center.....1
China Lake, CA 93555-6001
Attn: Dr. D. Burdick

Commanding General, Army Aviation Systems Command.....1
St. Louis, MO 63102

Director, Aeromechanics Laboratory.....1
U.S. Army (AVRADCOM)
NASA Ames Research Center
Moffett Field, CA 94035
Attn: Mr. D. Key

Director, NASA Ames Research Center.....1
Moffett Field, CA 94035
Attn: Dr. C. Jorgensen(MS 244-4,Code FII)